Advant® OCS
with Master software

PC Elements
Advant Controller 400 Series

Reference Manual

3BSE 002 418R601

Use of **DANGER**, **WARNING**, **CAUTION**, and **NOTE**

This publication includes, **DANGER**, **WARNING**, **CAUTION**, and **NOTE** information where appropriate to point out safety related or other important information.

DANGER	Hazards which could result in severe personal injury or death
WARNING	Hazards which could result in personal injury
CAUTION	Hazards which could result in equipment or property damage
NOTE	Alerts user to pertinent facts and conditions.

Although **DANGER** and **WARNING** hazards are related to personal injury, and **CAUTION** hazards are associated with equipment or property damage, it should be understood that operation of damaged equipment could, under certain operational conditions, result in degraded process performance leading to personal injury or death. Therefore, comply fully with all **DANGER**, **WARNING**, and **CAUTION** notices.

TRADEMARKS

Advant, AdvaCommand, AdvaInform, and AdvaBuild are registered trademarks of ABB Asea Brown Boveri Ltd, Switzerland.

Echelon, LON, LonTalk, LONWORKS, LONMARK, and Neuron are registered trademarks of Echelon Corporation.

NOTICE

The information in this document is subject to change without notice and should not be construed as a commitment by ABB Automation Products AB. ABB Automation Products AB assumes no responsibility for any errors that may appear in this document.

In no event shall ABB Automation Products AB be liable for direct, indirect, special, incidental or consequential damages of any nature or kind arising from the use of this document, nor shall ABB Automation Products AB be liable for incidental or consequential damages arising from use of any software or hardware described in this document.

This document and parts thereof must not be reproduced or copied without ABB Automation Products AB's written permission, and the contents thereof must not be imparted to a third party nor be used for any unauthorized purpose.

The software described in this document is furnished under a license and may be used, copied, or disclosed only in accordance with the terms of such license.

Copyright © ABB Industrial Systems AB 1999.

TABLE OF CONTENTS

Contents of Part 1(2)

Introduction

AMPL Language 1
Call 1
Call Name 1
Call Parameters 1
Graphic Symbol 2
Connection of PC Element Inputs and Outputs 2
Error Handling Integer Overflow 3
Overflow Real Numbers 3
Element Libraries 3
Programming with AMPL 3
Data Sheets 4
ELEMENT (C1, C2, ... Cn) 4
PC Elements According to Type 6
Short Form Description 13
Element Groups 21
Execution Times 22
Memory Space Requirements 32

Data Sheets

ABS 41
ADD 42
ADD-MR 43
ADD-MR1 44
ANALYSE 45
AND/OR 47
AND-O 49
BLOCK 51
COM-AIS 52
COM-CVI1 56
COM-CVO1 61
COM-DIS 66
COM-M 70
COM-MP51 74
COM-STAT 77
COMP 80
COMP-I 81
COMP-R 82

CONTENTS (continued)

Contents of Part 1(2)

CON-PU1 84
CONTRM 90
CONV 93
CONV-AI 94
CONV-BI 95
CONV-IA 98
CONV-IB 99
CONV-SA 102
COUNT 104
COUNT-DP 106
COUNT-L 108
CVB-I 110
CVB-O 114
DATE 120
DEMUX-MI 121
DEMUXA-M 123
DER 125
DISP 128
DISP-SEG 131
DIV 133
DIV-MR 134
DP820-I 135
DP820-O 138
DRI-CNV 141
DRI-R 142
DRI-S 144
DSP-R 146
DSP-S 148
EVENT 150
EXP 154
EXPAND 155
EXPAND-A 156
FAULT 158
FIFO 161
FIFO-RW 163
FILT-1P 166
FILT-2P 169
FPM-COM 172
FPM-I 174

CONTENTS (continued)

Contents of Part 1(2)

FPM-IA 176
FPM-O 177
FPM-OA 179
FREQ-MP 180
FREQ-SP 183
FUNCM 186
FUNG-1V 187
FUNG-2V 189
FUNG-T 191
FUZZYCON 194
GENBIN-I 201
GENBIN-O 203
GENCON-I 204
GENCON-O 206
GENUSD-I 207
GENUSD-O 209
INT 210
INV 212
KEYB-FU 213
KEYB-GR 215
KEYB-N1 218
LIM-N 222

Contents of Part 2(2)

LN 225
LON-R 226
LON-S 243
MAJ-R 246
MANSTN 247
MASTER 250
MAX 253
MED-R 255
MFB-IN 256
MFB-OUT 260
MIN 263
MMC-IND 265
MMC-ORD 267
MONO 268
MOTCON 270
MOVE 274
MOVE-A 275

CONTENTS (continued)

Contents of Part 2(2)

MUL 276
MUX-I 277
MUX-MI 279
MUX-MN 281
MUX-N 283
MUXA-I 285
MUXGE-MI 286
MUXGR-MI 288
NOVATUNE 290
NUM-IN 299
OR-A 301
OSC-B 303
OSC-SIN 304
OSC-SQW 305
P-1 307
PB-DIAG 309
PB-R 311
PB-S 313
P-DEADB 315
PCPGM 317
PCU-COM 319
PCU-I 320
PCU-O 322
PCU-SS 324
PDP 325
PI 328
PIDCON 331
PIDCONA 337
PIP 342
POS-A 345
POS-L 349
POS-O 351
PRINT 355
PULSE-S 358
RAMP 359
RAMP-S1 362
RATIOSTN 365
REG 368
REG-G 370

CONTENTS (continued)

Contents of Part 2(2)

REG-RET 373
REPORT 378
SCALE 380
SCALEDOS 385
SENDREQ 389
SEQ 391
SHIFT 397
SHIFT-L 399
SLAVEM 401
SQRT 402
SR 403
SR-AA, SR-AO, SR-OO, SR-OA 404
SR-D 406
STEP 407
SUB 414
SW 415
SW–C 416
TBL-R 417
TBL-RG 419
TBL–W 422
TBL-WG 424
TEXT 427
THRESH-L 429
TIME 430
TIMER 431
TON/ TOFF 434
TON-RET 435
TRIGG 436
VALVECON 437
XOR 440

Introduction

Advant Controller 400 Series is a range of products for control of various kind of industrial processes.

The system provides the ABB Master Programming Language methods, AMPL. AMPL is based on software implemented function blocks, **PC Elements**. The PC Elements are connected to each other into programs which form a complete control function. AMPL is used in all Advant Controllers and MasterPiece 200/1.

You use AMPL in a graphical environment with a personal computer and work station based engineering stations.

AMPL Language

The PC element is the smallest unit of AMPL. Each element performs a complete function, such as a logical AND gate, a timer, or a feedback control element.

Call

Each element has its own call name that enables it to be called regardless of the computer system the element is in. Many of the elements have call parameters which allow size, function, data type and other properties to be determined.

Call Name

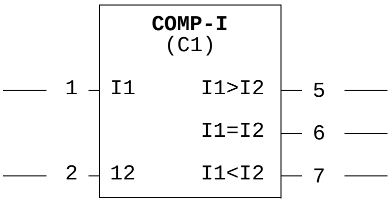
The mnemonic call name gives information about the basic function of the element. Further information about the function of the element, or the data types for which the element is intended, is given by a suffix to the name. For example, COMP-I, comparator for integers, and COMP-R, comparator for real numbers.

Call Parameters

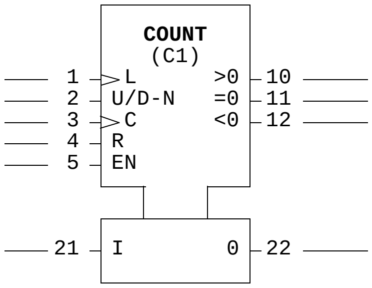
The call parameters for structure elements determine cyclicity, place in the cycle time table (determines the order of execution for PC modules having the same cyclicity) and, for a slave element, the identity of its master header. The call parameters for function elements may specify the size, the data type, the number of inputs and outputs etc.

Graphic Symbol

Each PC element has a graphic symbol similar to the figure below,



Normally, the call name of the element is written in the uppermost part of the symbol. For certain functions, for example, AND and OR, the name has been replaced by a symbol. When the element has superior control signals, the terminals for these signals are collected in a separate part of the symbol according to the figure below.



Connection of PC Element Inputs and Outputs

Only inputs and outputs with the same data type can be interconnected. If, for example, an output with data type I is to be connected to an input of type R, a code conversion element, for example, CONV (I, R), must be inserted in the data path.

Designation	**Data type**
I	Integer, 16 bit
IL	Integer, 32 bit
R	Real number
B	Boolean
T	Time
TR	Time, real representation
G	Group data, i.e. a collection of data with one of the above data types
A	Array

Error Handling Integer Overflow

The maximum range for integers is limited to the following values:

For I=–32768 .. 32767

For IL=–2147483648 .. 2147483647

When the range is exceeded, the output is set to the respective limiting value.

Overflow Real Numbers

The maximum range for real numbers is limited to -9.2×10^{18}... 9.2×10^{18}.

When the range is exceeded, the output is set to the respective limiting value. The smallest magnitude that can be represented by a real number is 5.0×10^{-20}. Any output with an absolute value smaller than this is set to 0.0.

Element Libraries

Each standard system has an element library which contains a defined subset of the total number of available PC elements. You will find the definition of the PC Element library under headline **Product Grouping**.

Programming with AMPL

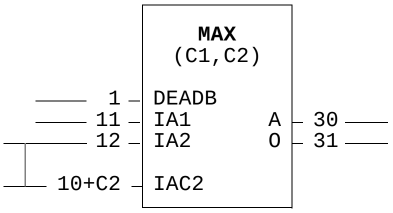
When you program a controller, choose PC elements from the library and include them in a draft PC diagram. The elements are then interconnected on the diagram, to describe the exchange of data between them.

The information on the draft PC program is used when entering the PC program in source code form on an engineering station. The engineering station translates the program into an object code, which is loaded into the target system.

Data Sheets

An element description starts with a short **Summary** that describes the main function.

To the right of the summary, the graphic symbol of the element is shown. Symbols with a variable number of connections are shown, as in the figure below.



Under the heading **Call**, the calling of the element is described.

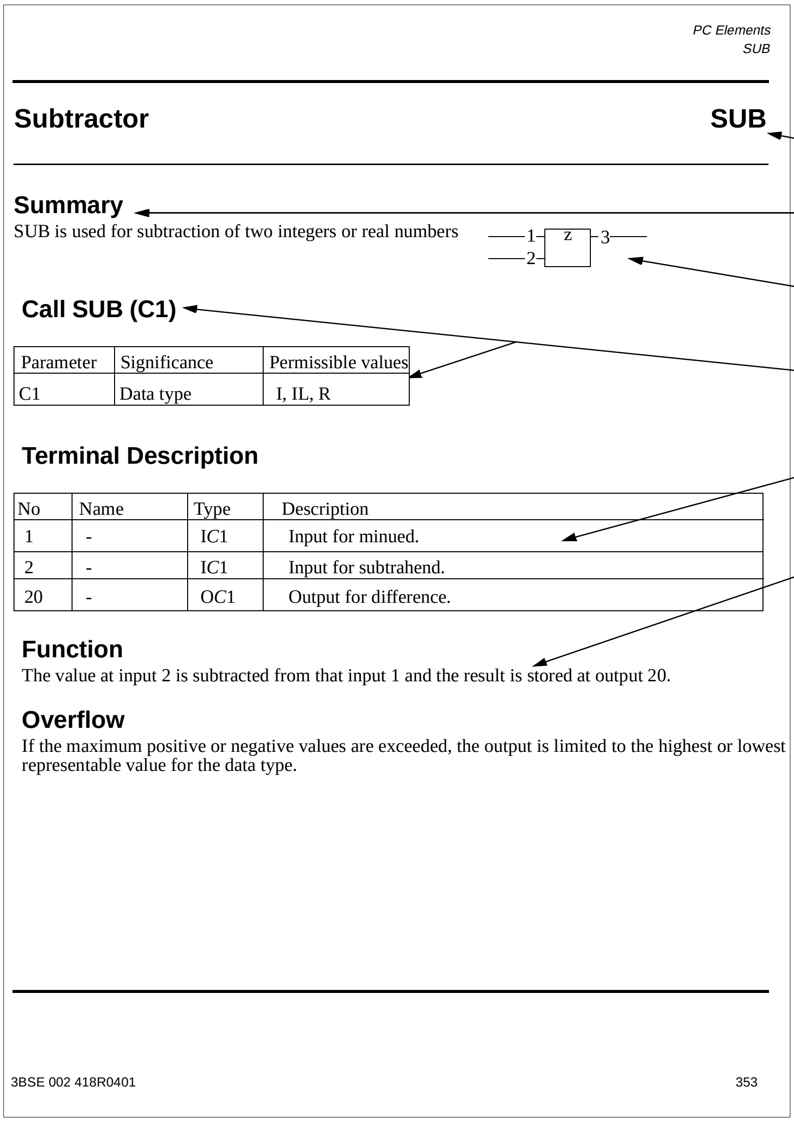
ELEMENT (C1, C2, ... C*n*)

If the element has call parameters, these are described in a table where their significance and permissible values are stated.

Element terminals are described under the heading **Connections**. This section is normally dominated by a table giving each terminal number, name, type and application.

The type of the connection is given as XY, where X is I or O if it is an input or an output terminal, Y indicates the data type according to the table under the heading **Terminal Description**.

The descriptions end with a detailed functional description under the heading **Function.**



Subtractor SUB

Summary

SUB is used for subtraction of two integers or real numbers

Call SUB (C1)

Parameter	Significance	Permissible values
C1	Data type	I, IL, R

Terminal Description

No	Name	Type	Description
1	-	IC1	Input for minued.
2	-	IC1	Input for subtrahend.
20	-	OC1	Output for difference.

Function

The value at input 2 is subtracted from that input 1 and the result is stored at output 20.

Overflow

If the maximum positive or negative values are exceeded, the output is limited to the highest or lowest representable value for the data type.

The element's call name.

Short description of the element's function.

The element's graphic symbol.

Call, with table over the call parameters, if any.

The element's terminals are described in a table.

Detailed description of the element's function.

Description, schematic outline

PC Elements According to Type

Element Type	**Element name**
Structure Elements	
Block Header	BLOCK
Control Module Header	CONTRM
Function Header	FUNCM
Master Header	MASTER
Program Header	PCPGM
Sequence Header	SEQ
Slave Header	SLAVEM
Step Header	STEP
Combinational Elements	
And Gate	AND
And Gate with Or inputs	AND-O
Inverter	INV
Or Gate	OR
Or Gate with And Inputs	OR-A
Threshold Function	THRESH-L
Exclusive Or Gate	XOR
Memory Elements	
Set/ Reset Memory	SR
Set/ Reset Memory with Data Input	SR-D
Set/ Reset Memory with And Inputs	SR-AA
Set/ Reset Memory with And/ Or Inputs	SR-AO
Set/ Reset Memory with Or/ And Inputs	SR-OA
Set/ Reset Memory with Or Inputs	SR-OO

Element Type	Element name
Time Elements	
Date Reading	DATE
Monostable	MONO
Oscillator	OSC-B
Oscillator for Sine Wave	OSC-SIN
Oscillator for Square Wave	OSC-SQW
Time Reading	TIME
Time Selector	TIMER
Time Delay Off	TOFF
Time Delay On	TON
Time Delay On with Retention	TON-RET
Trigger	TRIGG
Register Elements	
Array Demultiplexer	DEMUXA-M
Group Data Expander	EXPAND
Byte Array Data Expander	EXPAND-A
Queue Register	FIFO
Queue Register, Read/ Write	FIFO-RW
Data Copying	MOVE
Array Data Copying	MOVE-A
Register	REG
Group Data Register	REG-G
Register with Retention	REG-RET
Shift Register	SHIFT
Shift Register with Loading	SHIFT-L
Switch Elements	
Switch	SW
Changeover Switch	SW-C

Element Type	Element name
Multiplexer Elements	
Demultiplexer with Memory	DEMUX-MI
Array Demultiplexer	DEMUXA-M
Multiplexer with Integer Address	MUX-I
Multiplexer with Memory and Integer Address	MUX-MI
Multiplexer with Memory and 1 of N	MUX-MN
Multiplexer with 1 of N Address	MUX-N
Multiplexer of Group Data Elements	MUXGE-MI
Multiplexer of Group Data Vectors	MUXGR-MI
Code Converter Elements	
Code Converter	CONV
Code Converter, Array to Integer	CONV-AI
Code Converter, Boolean to Integer	CONV-BI
Code Converter, Integer to Array	CONV-IA
Code Converter, Integer to Boolean	CONV-IB
Code Converter, String to Array	CONV-SA
Arithmetic Elements	
Absolute Value	ABS
Adder	ADD
Adder with Multiplied Output	ADD-MR
Adder with Multiplied Output/ Inputs	ADD-MR1
Divider	DIV
Divider with Multiplied Inputs	DIV-MR
Exponentiation	EXP
Limiter	LIM-N
Natural Logarithm	LN
Majority Selector	MAJ-R
Median Selector	MED-R
Multiplier	MUL
Square Root	SQRT
Subtract	SUB

Element Type	**Element name**
Counter Elements	
Counter	COUNT
Counter with Limits	COUNT-L
Two-channel pulse counter	DP820-I, DP820-O
Comparator Elements	
Comparator	COMP
Comparator for Integer	COMP-I
Comparator for Real Numbers	COMP-R
Maximum Selector	MAX
Minimum Selector	MIN
Function Generators	
Function Generator with 1 Variable	FUNG-1V
Function Generator with 2 Variables	FUNG-2V
Function Generator with Time Variable	FUNG-T
Feedback and Filter Elements	
Three State Controller	CON-PU1
Derivator	DER
Filter with 1 Pole	FILT-1P
Filter with 2 Poles	FILT-2P
Fuzzy controller	FUZZYCON
Integrator	INT
P Function Proportional	P-1
Low Amplitude Filter	P-DEADB
PDP Function	PDP
PI-Regulator	PI
PIP-Function	PIP
Ramp Generator	RAMP
S-Ramp Generator	RAMP-S1
Adaptive Controller	NOVATUNE

Element Type	Element name
Panel Elements	
Numerical Display	DISP
Special Character Display	DISP-SEG
Function Keyboard	KEYB-FU
Group Function Keyboard	KEYB-GR
Numerical Keyboard	KEYB-N1
Numerical Input from Thumb Wheel Switch	NUM-IN
Data Set Elements	
Request Execution of Data Set	SENDREQ
Fieldbus Communication Elements	
Send commands to AIS in AC 100 Series.	COM-AIS
Send commands to DIS in AC 100 Series.	COM-DIS
Send commands to MB/MI/MIL/MR in AC 100 Series.	COM-M
Convert Drive data with scale factor	DRI-CNV
Receive drive data on Advant Fieldbus 100	DRI-R
Send drive data on Advant Fieldbus 100	DRI-S
Receive cyclic data on Advant Fieldbus 100	DSP-R
Send cyclic data on Advant Fieldbus 100	DSP-S
Receive data from a LONWORKS Network Variable	LON-R
Send data to a LONWORKS Network Variable	LON-S
Receive diagnostic data from PROFIBUS-DP slave	PB-DIAG
Receive data from PROFIBUS-DP slave	PB-R
Send data to PROFIBUS-DP slave	PB-S
MasterFieldbus Communication Elements	
Communication with MP 51	COM-MP51
Data from Convertor	COM-CVI1
Data to Convertor	COM-CVO1
Data from Advant Controller 110	MFB-IN
Data to Advant Controller 110	MFB-OUT

Element Type	Element name
Hardware Module Handling Elements	
Pulse Counter (DSDP 150)	COUNT-DP
Digital Speed Measurement (DSDC 111)	CVB-I
Digital Speed Control (DSDC 111)	CVB-O
Two-channel Pulse Counter	DP820-I, DP820-O
Free-Programmable Module, Commands (PU535)	FPM-COM
Free-Programmable Module, Input (PU535)	FPM-I
Free-Programmable Module, Input Array (PU535)	FPM-IA
Free-Programmable Module, Output (PU535)	FPM-O
Free-Programmable Module, Output Array (PU535	FPM-OA
Frequency Counter (DSDP 150)	FREQ-MP
Low Frequency Counter (DSDP 150)	FREQ-SP
Pulse Counter, Common Control (DSDP 170)	PCU-COM
Pulse Counter, Input (DSDP 170)	PCU-I
Pulse Counter, Output (DSDP 170)	PCU-O
Pulse Counter, Simultaneous Sampling (DSDP 170)	PCU-SS
Positioning, Analog Control (DSDP 140)	POS-A
Length Measurement (DSDP 140)	POS-L
Positioning, On -Off Control (DSDP 140)	POS-O
Synchronization of Pulse Counter (DSDP 110)	PULSE-S
Weighing (DSXW 110	SCALE
Batching (DSXW 110)	SCALEDOS
Supervision Elements	
System Load Measurement	ANALYSE
System Supervision	COM-STAT
Application Adapted Elements	
Error Signal Handling	FAULT

Element Type	Element name
Text Handling Elements	
Event Sending	EVENT
Multiplexer for Text	MUXA-I
Print Request	PRINT
Report Printout	REPORT
Text Line	TEXT
Functional Unit Elements	
Binary Functional Unit Inputs	GENBIN-I
Binary Functional Unit Outputs	GENBIN-O
Analog Functional Unit Inputs	GENCON-I
Analog Functional Unit Outputs	GENCON-O
User Defined Functional Unit Inputs	GENUSD-I
User Defined Functional Unit Outputs	GENUSD-O
Manual Station	MANSTN
Operator Station Indications	MMC-IND
Operator Station Orders	MMC-ORD
Motor Controller	MOTCON
Process Controller	PIDCON
Process Controller with Auto Tuning	PIDCONA
Ratio Station	RATIOSTN
Valve Controller	VALVECON
Table Handling Elements	
Read Table	TBL-R
Read Table Row	TBL-RG
Write Table	TBL-W
Write Table Row	TBL-WG

Short Form Description

Element	Description
ABS	Calculates the absolute value of an integer or a real number. The result is multiplied by a scale factor.
ADD	Adds up to 19 integers or real numbers.
ADD-MR	Adds two groups of up to 19 real numbers each and subtracts the second from the first. The result is multiplied by a real number.
ADD-MR1	Adds up to nine products of real numbers. The result is multiplied by a real number.
ANALYSE	Measures the load on the system and gives alarms for overload and at optional limits.
AND	Gives the Boolean product of up to 19 Boolean values.
AND-O	Gives the Boolean product of up to 9 Boolean values and up to 5 Boolean sums of a maximum of 9 Boolean values each.
BLOCK	Module header for blocks.
COM-AIS	Used for sending commands to DB element AIS objects in Advant Controller 100 Series on Advant Fieldbus 100.
COM-CVI1	Used as a communication link between a PC program and a converter TYRAK L. Fetches information from the convertor.
COM-CVO1	Used as a communication link between a PC program and a converter TYRAK L. Gives set points to the converter.
COM-DIS	Used for sending commands to DB element DIS objects in Advant Controller 100 Series on Advant Fieldbus 100.
COM-M	Used for sending commands of boolean, integer, long integer or real type values to MB/MI/MIL/MR objects in Advant Controller 100 Series on Advant Fieldbus 100.
COM-MP51	Communication with MasterPiece 51 equipment.
COM-STAT	Supervision of networks.
COMP	Compares two values of optional type.
COMP-I	Used to compare two integers.
COMP-R	Used to compare two real numbers.
CON-PU1	Three state controller with outputs for increase/decrease for control via an integrating actuator.
CONTRM	Module header for control modules.
CONV	Conversion of data between the types I, IL, R, T and TR.
CONV-AI	Converts four characters to a long integer (IL).
CONV-BI	Converts data in BC, BCD, 1-of-N code, Gray code, and 16/32 bit packing, given at a number of Boolean inputs into data in integer form (I or IL).

Element	Description
CONV-IA	Converts a long integer to four characters.
CONV-IB	Converts data in integer form (I or IL) into data in BC, BCD, 1-of-N code and uncoded unpacking at a number of Boolean inputs.
CONV-SA	Converts string to array
COUNT	Preset able counter for counting pulses, up or down. With checking of the value relative to 0.
COUNT-DP	Used with DSDP 150 for pulse counting with frequencies up to 10 kHz.
COUNT-L	Preset able counter for counting pulses, up or down. With checking of the value relative to 0 and relative to optional max and min values.
CVB-I	Used to read measured values, indications and error signals from the digital speed measurement and speed control board DSDC 111.
CVB-O	Used to write setpoints, orders and parameters to the digital speed measurement and speed control board DSDC 111.
DATE	Reads the current date.
DEMUX-MI	Demultiplexer with memory and integer address.
DEMUXA-M	Demultiplexer (Array) with memory and integer address.
DER	Used to give derivation effect. The element has inputs and outputs for limit values, setting of gain, filter function for restricting the derivation effect and the possibility of following and external reference.
DISP	Used to show values of one of the types B, I, IL, R, T, TR on the numerical display DSHM 110 or DSHM 120.
DISP-SEG	Used to display special characters on numerical display DSHM 110 or DSHM 120.
DIV	Used for dividing two real numbers or two integers. For integers the quotient is obtained along with a remainder at a special output.
DIV-MR	Used to divide one product of real numbers with another.
DRI-CNV	DRI-CNV (DRIve CoNVert) converts data to from integer and real by using a scaling algorithm.
DRI-R	DRI-R (DRIve - Receive) is used for receiving drive cyclic data on Advant Fieldbus 100.
DRI-S	DRI-S (DRIve - Send) is used for sending drive cyclic data on Advant Fieldbus 100.
DP820-I	Used together with DP820-O to read and control the advanced pulse counter board DP820. DP820-I is used to read measured values from the board.
DP820-O	Used together with DP820-I to read and control the advanced pulse counter board DP820. DP820-OI is used to give setpoints, orders and parameters to the channels on the board.
DSP-R	DSP-R (Data Set Peripheral -Receive) is used to receive cyclic data on Advant Fieldbus 100.

Element	Description
DSP-S	DSP-S (Data Set Peripheral -Send) is used for sending of cyclic data on Advant Fieldbus 100.
EVENT	Used to transmit events.
EXP	Calculates the exponential of a real number. The result is multiplied by a scale factor.
EXPAND	Used to expand the values in group data to normal data types.
EXPAND-A	Used to extract data from a byte array.
FAULT	Used for fault indication with blinking lights with up to 9 fault signal points and one common output for group wise fault indication.
FIFO	Queue register with up to 9 queues each with a maximum of 64 places.
FIFO-RW	Queue register with up to 9 queues each with a maximum of 64 places. Data may be read, written or deleted at optional queue places.
FILT-1P	1-pole low-pass filter with options for limiting the output signal and for following an external reference.
FILT-2P	2-pole low-pass filter with options for limiting the output signal and for following an external reference.
FPM-COM	Used to control the execution of the application program and to receive status information from the programmable board PU535.
FPM-I	Used to read data from the programmable module PU535.
FPM-IA	Used to read a byte array of 80 characters from the programmable module PU535.
FPM-O	Used to write data to the programmable module PU535.
FPM-OA	Used to write a byte array of 80 characters to the programmable module PU535.
FREQ-MP	Used with DSDP 150 for measuring frequencies up to 10 kHz.
FREQ-SP	Used with DSDP 150 for measuring frequencies.
FUNCM	Module header for function module.
FUNG-1V	Generates a piece-wise linear function of one variable using tables, with up to 255 elements. If the function is strictly monotonous the inverse of the function can be generated by balancing.
FUNG-2V	Generates a function of two variables using tables with up to 63 elements.
FUNG-T	Generates a piece-wise linear function of a time variable using tables with up to 255 elements. If the function is strictly monotonically the inverse of the function may be generated with balancing.
FUZZYCON	Used as a Fuzzy Controller, a rule based control strategy. Supports multiple inputs as well as multiple outputs.
GENBIN-I	Used in the functional unit GENBIN, see the manuals for ABB Master Functional Units.

Element	Description
GENBIN-O	Used in the functional unit GENBIN, see the manuals for ABB Master Functional Units.
GENCON-I	Used in the functional unit GENCON, see the manuals for ABB Master Functional Units.
GENCON-O	Used in the functional unit GENCON, see the manuals for ABB Master Functional Units.
GENUSD-I	Used in the functional unit GENUSD, see the manuals for ABB Master Functional Units.
GENUSD-O	Used in the functional unit GENUSD, see the manuals for ABB Master Functional Units.
INT	Used to give integration effect. The element has inputs and outputs for limit values, setting of gain and following with an external reference.
INV	Used to invert a Boolean value.
KEYB-FU	Used to connect function keyboard DSHM 130 to a PC system.
KEYB-GR	Used to communicate with the function keyboard DSHM 130 when its keys are to be divided into groups.
KEYB-N1	Used for numerical entry from a numeric keyboard DSHM 120.
LIM-N	Used for limiting integers, real numbers or time values. Several limit values may be selected.
LN	Calculates the natural logarithm of a real number. The result is multiplied by a scale factor.
LON-R	Used for receiving data from a LONWORKS Network Variable
LON-S	Used for sending data to a LONWORKS Network Variable
MAJ-R	Used to calculate the average of 3 or 4 numbers, but with exclusion of an input that deviates from the calculated average with more than a given amount.
MANSTN	Used in the functional unit MANSTN, see the manuals for ABB Master Functional Units.
MASTER	Module header for master modules.
MAX	Selects the largest of up to 19 integers or real numbers.
MED-R	Calculates the median of up to 19 real numbers.
MFB-IN	The element MFB-IN is used to transfer data to an Advant Controller 410/450 from an Advant Controller 110 connected via MasterFieldbus.
MFB-OUT	The element MFB-OUT is used to transfer data from an Advant Controller 410/450 to an Advant Controller 110 connected via MasterFieldbus.
MIN	Selects the smallest of up to 19 integers or real numbers.
MMC-IND	Used in the functional unit MMC, see the manuals for ABB Master Functional Units.
MMC-ORD	Used in the functional unit MMC, see the manuals for ABB Master Functional Units.
MONO	Used to generate a pulse with a given duration when the input changes from 0 to 1. The pulse duration is specified at an input. Retriggability is controlled with an input.
MOTCON	Used in the functional unit MOTCON, see the manuals for ABB Master Functional Units.

Element	Description
MOVE	Used to move data, for example from one place in the database to another.
MOVE-A	Used to move an array from one place in the database to another.
MUL	Used for multiplication of up to 19 integers or real numbers.
MUX-I	Selector for up to 19 inputs. Integer address.
MUX-MI	Selector for up to 19 inputs. Integer address and memory.
MUX-MN	Selector for up to 19 inputs. 1-of-N address and memory.
MUX-N	Selector for up to 19 inputs. 1-of-N address.
MUXA-I	Selector for up to 19 inputs with text arrays with a maximum length of 80 characters. Integer address.
MUXGE-MI	Selects one element from a group data.
MUXGR-MI	Selector for group data.
NOVATUNE	Self-adjusting adaptive controller.
NUM-IN	Used for the acquisition of values from thumb wheel unit DSHM 140.
OR	Gives the Boolean sum of up to 19 Boolean values.
OR-A	Gives the Boolean sum of up to 9 Boolean values and up to 5 Boolean products each of a maximum of 9 Boolean values.
OSC-B	Oscillator for a Boolean variable with variable frequency and pulse time.
OSC-SIN	Oscillator for sine wave with inputs for setting period time and amplitude.
OSC-SQW	Oscillator for square wave where each period may contain up to 4 time intervals with a given amplitude and duration.
P-1	Used as an amplifier in feedback systems. The element permits output signal limiting and following.
PB-DIAG	Used for access of diagnostic data from a PROFIBUS-DP slave.
PB-R	Used for access of input data from a PROFIBUS-DP slave.
PB-S	Used for sending data to PROFIBUS-DP slave.
P-DEADB	Used as a filter for signal changes less than a given dead band. The output signal may be limited.
PCPGM	Module header for program.
PCU-COM	Used together with PCU-I, PCU-O and PCU-SS to read and control the advanced pulse counter board DSDP 170. PCU-COM is used for initiation and supervision of DSDP 170 board.

Element	Description
PCU-I	Used together with PCU-O, PCU-COM and PCU-SS to read and control the advanced pulse counter board DSDP 170. PCU-I is used to read measured values individually for one of the four transducer channels on the board.
PCU-O	Used together with PCU-I, PCU-COM and PCU-SS to read and control the advanced pulse counter board DSDP 170. PCU-O is used to give setpoints, orders and parameters individually for one of the four transducer channels on the board.
PCU-SS	Used together with PCU-I, PCU-O and PCU-COM to read and control the advanced pulse counter board DSDP 170. PCU-SS is used when simultaneous sampling of several channels for up to five boards is required.
PDP	Used to give proportional effect and limited derivation effect. The element permits limiting of the output signal and following.
PI	Used as a standard PI regulator for serial compensation in feedback systems. The element permits limiting of the output signal and following.
PIDCON	Used in the functional unit PIDCON, see the manuals for ABB Master Functional Units.
PIDCONA	Used in the functional unit PIDCONA, see the manuals for ABB Master Functional Units.
PIP	Used to give proportional effect and limited integration effect. The element permits limiting of the output signal and following.
POS-A	Used with DSDP 140 for length measurement or positioning.
POS-L	Used with DSDP 140 for length measurement with a digital output for coincidence.
POS-O	Used with DSDP 140 for positioning with a digital output signal.
PRINT	Element for output a printer, locally connected or via a network.
PULSE-S	Used to synchronize pulse counters on a pulse counter board,e.g. DSDP 110.
RAMP	Used to limit the rate of change of a signal. The element permits limiting of the output signal and following.
RAMP-S1	Used to generate a ramp signal function in which the transitions between constant outward signal and ramp are soft.
RATIOSTN	Used in the functional unit RATIOSTN, see the manuals for ABB Master Functional Units.
REG	Used as a memory element with an optional number of positions.
REG-G	Used to assemble simple variables into one single variable of group type.
REG-RET	Used as a memory element with an optional number of positions. The element has functions for controlling the value of the outputs after a power failure.
REPORT	Used to request the printing of reports.
SCALE	Used for weighing.
SCALEDOS	Used for dosing.

Element	Description
SENDREQ	Used to request event control of a data set element.
SEQ	Module header for a sequence.
SHIFT	Used as a shift register with optional length.
SHIFT-L	Used as a shift register with optional length. All positions may be loaded and read.
SLAVEM	Module header for a slave. Subordinate to a MASTER.
SQRT	Calculates the square root of a real number. The result is multiplied by a scale factor.
SR	Memory for Boolean variables. The element has one Set and one Reset input.
SR-AA	Memory for Boolean variables. The element has And gates with up to 9 inputs on the Set and Reset inputs.
SR-AO	Memory for Boolean variables. The element has one And gate with up to 9 inputs on the Set input and one Or gate with up to 9 inputs on the Reset input.
SR-D	Memory for Boolean variables. The element has one Set and one Reset input and data and clock inputs for clocking in data.
SR-OA	Memory for Boolean variables. The element has one Or gate with up to 9 inputs on the Set input and one And gate with up to 9 inputs on the Reset input.
SR-OO	Memory for Boolean variables. The element has Or gates with up to 9 inputs on the Set and Reset inputs.
STEP	Module header for a step. Subordinate to an SEQ.
SUB	Subtraction of one real or one integer number from another.
SW	Switching element with up to 9 closing channels.
SW-C	Switching elements with up to 9 change over channels.
TBL-R	Used to get data from database.
TBL-RG	Used to read values from one stage in the database.
TBL-W	Used to write data to database.
TBL-WG	Used to write data to one stage in the database.
TEXT	Used to assemble a text line of up to 72 characters from up to 8 sub texts.
THRESH-L	Used to determine when more than a given number of Boolean signals are set to 1.
TIME	Used to read the current system time.
TIMER	Used to generate pulses at up to 9 outputs. The pulse duration is determined for each output by to times within a day.
TOFF	Delay for turning a Boolean signal off.
TON	Delay for turning a Boolean signal on.
TON-RET	Delay for turning a Boolean signal on with accumulating time measurement.

Element	Description
TRIGG	Provides a pulse during one program cycle when the input signal changes from 0 to 1.
VALVECON	Used in the functional unit VALVECON, see the manuals for ABB Master Functional Units.
XOR	Used to give a signal when one and only one of the input signals is true.

Element Groups

The elements belong to the following software packages.

Advant Controller 410 with QC01-BAS11,
Advant Controller 450 with QC07-BAS41 (previously QC02-BAS21):

ABS, ADD, ADD-MR, ADD-MR1, ANALYSE, AND, AND-O, BLOCK,

COM-CVI1, COM-CVO1, COM-MP51, COM-STAT, COMP, COMP-I, COMP-R, CONTRM, CONV, CONV-AI, CONV-BI, CONV-IA, CONV-IB, CONV-SA, COUNT, COUNT-DP, COUNT-L, DATE, DEMUX-MI, DEMUXA-M, DISP, DISP-SEG, DIV, DIV-MR, DP820-I, DP820-O, DRI-CNV, DRI-R, DRI-S, DSP-R, DSP-S,

EVENT, EXPAND, EXPAND-A, FAULT, FIFO, FIFO-RW, FPM-COM, FPM-I, FPM-O, FPM-IA, FPM-OA, FREQ-MP, FREQ-SP, FUNCM,

GENBIN-I, GENBIN-O, GENCON-I, GENCON-O, GENUSD-I, GENUSD-O, INV,

KEYB-FU, KEYB-GR, KEYB-N1, LIM-N, LON-R, LON-S,

MASTER, MAX, MFB-IN, MFB-OUT, MIN, MMC-IND, MMC-ORD, MONO, MOTCON, MOVE, MOVE-A, MUL, MUX-I, MUX-MI, MUX-MN, MUX-N, MUXA-I, NUM-IN,

OR, OR-A, OSC-B, PB-DIAG, PB-R, PB-S, PCPGM, PCU-COM, PCU-I, PCU-O, PCU-SS, POS-A, POS-L, POS-O, PRINT, PULSE-S,

RAMP-S1, REG, REG-G, REG-RET, REPORT, SCALE, SCALEDOS, SENDREQ, SEQ, SHIFT, SHIFT-L, SLAVEM, SQRT, SR, SR-AA, SR-AO, SR-D, SR-OA, SR-OO, STEP, SUB, SW, SW-C,

TBL-R, TBL-RG, TBL-W, TBL-WG, TEXT, TIME, TIMER, TOFF, TON, TON-RET, TRIGG, VALVECON, XOR.

Advant Controller 410 with QC01-LIB11,
Advant Controller 450 with QC07-LIB41 (previously QC02-LIB21):

CON-PU1, CVB-I, CVB-O, DER, EXP, FILT-1P, FILT-2P, FUNG-1V, FUNG-2V, FUNG-T, INT, LN, MAJ-R, MED-R, MUXGE-MI, MUXGR-MI, OSC-SIN, OSC-SQW, P-1, P-DEADB, PDP, PI, PIP, RAMP, THRESH-L.

Advant Controller 410 with QC01-LIB12,
Advant Controller 450 with QC07-LIB42 (previously QC02-LIB22):

MANSTN, NOVATUNE, PIDCON, PIDCONA (Advant Controller 450 only), RATIOSTN.

Advant Controller 450 with QC07-COM41:

COM-AIS, COM-DIS, COM-M.

Advant Controller 450 with QC07-FUZ41:

FUZZYCON

Execution Times

Calculation

The execution times, specified in the table below, apply for Advant Controller 410 and Advant Controller 450.

When the execution times have been calculated for all PC elements, a general increment is to be added for calculation of the total execution time. This addition is to be made for each connection between PC elements, connected via read-in or read-out elements. Such connections are created when PC elements under different interpreters are interconnected and when PC elements are connected to Data Base elements.

Where read in and read out elements are created, is indicated in the PC diagram and PC list. In these, the read in elements are marked with G (Get) and read out elements with P (Put).
The time to be added to the total time is 6 μs, for Advant Controller 410, and 2μs, for Advant Controller 450, per read in and read out element. This time increases if the read-out is connected to a DOS or AOS Data Base element and the value has changed since last read out.
For more information about execution times for I/O, see Advant Controller 410 and Advant Controller 450 User's Guide.

Name	AC 410 Execution Time, typical (μs)	AC 450, Execution Time, typical (μs)	Notes
ABS	11 13 27	3.7 4.3 9	I IL R
ADD	6+2×C2 11+13×C2	2+0.6×C2 3.7+4.3×C2	C1=I, IL C1=R
ADD-MR	40+13(C1+C2)	13+4(C1+C2)	
ADD-MR1	10+30×C1	3+10×C1	
ANALYSE	15+23×C1+7×C2	5+7.5×C1+2×C2	
AND	4+2×q	1.3+0.6×q	q=number of inputs that must be tested
AND-O	10+2×q	3+0.6×q	q=number of inputs that must be tested
BLOCK	9	3	
COM-AIS	-	70	
COM-CVI1	26+60×C1+15×C2+33×q	10+20×C1+5×C2+11×q	q=number of scaled values
COM-CVO1	80+100×C1+9×q	26+35×C1+3×q	q=number of scaled values
COM-DIS	-	70	
COM-M	-	70	
COM-MP51	32+2×C1+3×C2×C3	11+0.6×C1+1×C2×C3	
COM-STAT	150+80×C1+1180×q	50+27×C1+395×q	q=the total number of nodes

Name	AC 410 Execution Time, typical (µs)	AC 450, Execution Time, typical (µs)	Notes
COMP	14 17+1.5×q	5 6+1.5×q	C1=B, I, IL, T, R, TR C1=A, q=number of equal characters at the start of the string
COMP-I	10	3	C1=I, IL
COMP-R	24+4(C1+C2)	8+1(C1+C2)	
CON-PU1	**C1=0:** 40 180 315 **C1=1:** 50 200 620	**C1=0:** 13 60 105 **C1=1:** 16 66 205	Stable At pulse time typical At pulse time max Stable At pulse time typical At pulse time max **Note:** High value of TMBC essentially increases the load
CONTRM	130	45	
CONV	13, 14 15, 17 31, 24 29, 25 35, 22 33, 24 14, 18 48, 50 13, 11 55, 47	4, 4 5, 5 10, 8 10, 8 11, 7 11, 8 4, 6 16, 17 4, 4 18, 15	I→IL, IL→I I→T, T→I I→R, R→I I–TR, TR→I IL→R, R→IL IL→TR, TR→IL IL→T, T→IL R→T, T→R R→TR, TR→R T→TR, TR→T
CONV-AI	17	6	
CONV-BI	36+3×C3 28+4×C3 25+2×C3	12+1×C3 9+1.3×C3 8+0.7×C3	C2=2 (BCD) C2=4 (Gray-code) others
CONV-IA	20	7	
CONV-IB	13	4	
CONV-SA	620 18	210 6	Active Not active
COUNT	20	7	
COUNT-DP	56	19	
COUNT-L	31	10	

Name	AC 410 Execution Time, typical (μs)	AC 450, Execution Time, typical (μs)	Notes
CVB-I	330	110	Calculated value
CVB-O	350	116	Calculated value
DATE	60	20	
DEMUX-MI	38	13	
DEMUXA-M	97+0.9×C1 97+2.2×C2+0.6×C1×C2	33+0.3×C1 33+0.7×C2+0.2×C1×C2	A>0 A=0
DER	116	40	
DISP	400 460 180 540 690	130 160 60 180 230	C1=I C1=IL C1=B C1=R, TR C1=T
DISP-SEG	120	40	
DIV	12 16 32	3 4 11	C1=I C1=IL C1=R
DIV-MR	4+18 (C1+C2)	2+6 (C1+C2)	
DP820-I	300	100	
DP820-O	300	100	
DRI-CNV	50	15	
DRI-R	140 + 170×C1	45 + 55×C1	
DRI-S	50 + 170×C1	15 + 55×C1	
DSP-R	180	60	
DSP-S	180	60	
EVENT	20+3×C1	7+1×C1	Idle
EXP	200	76	
EXPAND	34+1×C3+2×C4	11+0.4×C3+0.7×C4	
EXPAND-A	12 41 46+3×LEN 104+5×LEN	4 14 15+1×LEN 35+1.6×LEN	S=0 S=1 C1=A S=1 C1=I, IL S=1 C1=R LEN is the value at input terminal LEN
FAULT	20+14×C1	7+5×C1	

Name	**AC 410 Execution Time, typical (µs)**	**AC 450, Execution Time, typical (µs)**	**Notes**
FIFO	23	8	
FIFO-RW	39+2×C2	13+0.7×C2	
FILT-1P	114	38	
FILT-2P	189	63	
FPM-COM	40+2.5×C1	12+1×C1	
FPM-I	30	10	
FPM-IA	35	12	
FPM-O	45+4×C2+6(C4+C6+C8)	15+1.5×C2+2(C4+C6+C8)	
FPM-OA	36+2×C1	12+0.7×C1	
FREQ-MP	42	14	
FREQ-SP	73 150	25 50	Typical Max
FUNCM	17	6	
FUNG-1V	90+0.5×C1	30+0.2×C1	
FUNG-2V	153	50	
FUNG-T	135	45	
FUZZYCON	Not valid for AC 410	455+4x(C1+C2+C5+C6)+ 1.4xC2xC3+(70xC4+A)+ 1.2xC6xC7+(50xC8+B) 455+4x(C1+C2+C5+C6)+ 1.4xC2xC3+(70xC4+A)+ 1.2xC6xC7+(40xC8+B)	Controlled Variables in AUTO mode and, Output Variables in AUTO or E3(1) mode.(2) Controlled Variables in BAL mode and Output Variables in MAN mode. A=30 if C4 <> 0 else A=0. B=30 if C8 <> 0 else B=0.
GENBIN-I	240	80	
GENBIN-O	38	13	
GENCON-I	100	35	
GENCON-O	19	6	
GENUSD-I	109	36	
GENUSD-O	49	16	
INT	78	26	
INV	8	2.5	

Name	**AC 410 Execution Time, typical (μs)**	**AC 450, Execution Time, typical (μs)**	**Notes**
KEYB-FU	280	95	
KEYB-GR	140+140×C1	50+50×C1	
KEYB-N1	410	140	C3=I
	480	160	C3=IL, idle
	200	70	C3=B, idle
	550	190	C3=R, idle
	540	180	C3=TR, idle
	640	220	C3=T, idle
LIM-N	20+2×C2	7+1×C2	C1=I, IL, T
	28+2×C2	9+1×C2	C1=R, TR
LN	211	70	
LON-R	31.2 + 2.4×C2	10.4 + 0.8×C2	When ENTEST = 1, else
	48	16	When LONTYPE = 99 else
	39+ 3xC2 + 9×C4	13 + 1xC2 + 3×C4	C1 = B, LONTYPE = US, UL
	39 + 6×C2 + 9×C4	13 + 2×C2 + 3×C4	C1 = B, LONTYPE = FT
	42+ 1.5×C2 + 9×C4	14 + 0.5×C2 + 3×C4	C1 = B, LONTYPE = <XY>
	42+ 4.8×C2 + 9×C4	14 + 1.6×C2 + 3×C4	C1 = I, LONTYPE = US
	42+ 5.4×C2 + 9×C4	14 + 1.8×C2 + 3×C4	C1 = I, LONTYPE = SS
	36+ 6.6×C2 + 9×C4	12 + 2.2×C2 + 3×C4	C1 = I, LONTYPE = SL
	42+ 5.4×C2 + 9×C4	14 + 1.8×C2 + 3×C4	C1 = I, LONTYPE = FT
	45+ 9.6×C2 + 9×C4	15 + 3.2×C2 + 3×C4	C1 = I, LONTYPE = <XY>
	36+ 6.6×C2 + 9×C4	12 + 2.2×C2 + 3×C4	C1 = IL, LONTYPE = UL
	36+ 7.2×C2 + 9×C4	12 + 2.4×C2 + 3×C4	C1 = IL, LONTYPE = SS
	36+ 7.5×C2 + 9×C4	12 + 2.5×C2 + 3×C4	C1 = IL, LONTYPE = UL,SL
	42+ 12×C2 + 9×C4	14 + 4.0×C2 + 3×C4	C1 = IL, LONTYPE = <XY>
	36+ 36×C2 + 9×C4	12 + 12×C2 + 3×C4	C1 = IL, LONTYPE = FP
	42+ 8.1×C2 + 9×C4	14 + 2.7×C2 + 3×C4	C1 = IL, LONTYPE = S3
	36+ 27.9×C2 + 9×C4	12 + 9.3×C2 + 3×C4	C1 = R, LONTYPE = SS,UL,SL
	36+ 28.8×C2 + 9×C4	12 + 9.6×C2 + 3×C4	C1 = R, LONTYPE = S3
	39+ 8.7×C2 + 9×C4	13 + 2.9×C2 + 3×C4	C1 = R, LONTYPE = FP
	36+ 51×C2 + 9×C4	12 + 17×C2 + 3×C4	C1 = T, LONTYPE = SR
	39+ 8.7×C2xC3 + 9×C4	13 + 2.9×C2xC3 + 3×C4	C1 = A, LONTYPE = SR

Name	AC 410 Execution Time, typical (µs)	AC 450, Execution Time, typical (µs)	Notes
LON-S	30 58+ 3×C2 69+ 3×C2 63+ 3×C2 60+ 4.5×C2 60+ 4.5×C2 66+ 30×C2 54+ 7.5×C2 57+ 39×C2 63+ 3×C2 63+ 9×C2 60+ 3×C2xC3	10 26+ 1×C2 23 + 1×C2 21+ 1×C2 20+ 1.5×C2 20+ 1.5×C2 22 + 10×C2 18 + 2.5×C2 19 + 13×C2 21+ 1×C2 21+ 3×C2 20+ 1×C2xC3	When LONTYPE = 99, else C1 = B, LONTYPE = US,UL,FT C1 = B, LONTYPE = <XY> C1 = I, LONTYPE = US,SS,SL,FT C1 = I, LONTYPE = <XY> C1=IL, LONTYPE = US,SS,UL,SL,<XY> C1 = IL, LONTYPE = FP C1 = IL, LONTYPE = S3 C1 = R, LONTYPE = SS,UL,SL,S3 C1 = R, LONTYPE = FP C1 = T, LONTYPE = SR C1 = A, LONTYPE = SR
MAJ-R	68+45×C1	23+15×C1	
MANSTN	260+25×C1+35×I	86+8×C1+12×I	I=1 when OUT is in limit, else 0
MASTER	65	22	
MAX	10+3×C2 13+3×C2 13+4×C2	3+1×C2 4+1×C2 4+1×C2	C1=I C1=IL C1=R
MED-R	33+1.3×C1×C1 60+1.3×C1×C1	11+0.4×C1×C1 20+0.4×C1×C1	C1 odd C1 even
MFB-IN	90+1.5(C1+C2+C3 +C4+C5)+1×C6	30+0.5(C1+C2+C3 +C4+C5)+0.3×C6	
MFB-OUT	100+3(C1+C2+C3 +C4+C5)+1.5×C6	35+1(C1+C2+C3 +C4+C5)+0.5×C6	
MIN	12+3×C2 15+3×C2 16+4×C2	4+1×C2 5+1×C2 5+1×C2	C1=I C1=IL C1=R
MMC-IND	110+4×C1+3×C2	35+1.5×C1+1×C2	
MMC-ORD	25+4×C1	8+1×C1	
MONO	13	5	
MOTCON	515 735 770	170 245 345	C1 ... C6=0 C1 ... C5=1, C6=0 C1 ... C6=1
MOVE	9+3×C2 7+2×C2 9+4×C2	3+1×C2 2.5+0.7×C2 3+1.3×C2	C1=B C1=I,IL,R C1=T,TR
MOVE-A	5+2.9×C2+0.6×C1×C2	1.7×1×C2+0.2×C1×C2	

Name	AC 410 Execution Time, typical (µs)	AC 450, Execution Time, typical (µs)	Notes
MUL	14 10+1×C2 25+7×C2	5 3+0.3×C2 8+2×C2	C1=I C1=IL, 2 < C2 < 19 C1=R
MUX-I	12	4	C1=B=I=IL=R=TR=T
MUX-MI	19	6	C1=B=I=IL=R=T=TR
MUX-MN	22+2×C2	7+0.7×C2	C1=B=I=IL=R=T=TR
MUX-N	13+2×C2	4+0.7×C2	C1=B=I=IL=R=T=TR
MUXA-I	70+0.2×C2	23+0.1×C2	
MUXGE-MI	23	8	
MUXGR-MI	19+0.7×C2 19+1.1×C2 7+1.1×C2 19+0.8×C2	6+0.2×C2 6+0.3×C2 2+0.4×C2 6+0.3×C2	C1=I C1=IL, T C1=R, TR C1=B
NOVATUNE	670+T_{reg}+T_{adapt} T_{reg}=(370+55×N_{par}) / T T_{adapt}=(1800+135×N_{par}^2) / TA	220+T_{reg}+T_{adapt} T_{reg}=(120+18×$N_{par)}$ / T T_{adapt}=(600+45×N_{par}^2) / TA	T_{reg} is regular time when AUTO=1 T_{adapt} is adaptor time when ADAPON=1 T is terminal 22, T TA is terminal 23, TA N_{par} is the sum of the function parameters F1, F2 and F3, i.e. N_{par}=(NA+NB+NC)
NUM-IN	140	50	
OR	4+2×q	1.3+0.6×q	q=number of tested inputs
OR-A	6+2×q	2+0.6×q	q=number of tested inputs
OSC-B	13	4	
OSC-SIN	94	31	
OSC-SQW	16	5	
P-1	67 48	22 16	BAL=0 BAL=1
PB-DIAG	300	100	
PB-R	300	100	
PB-S	300	100	
P-DEADB	46	16	
PCPGM	66	22	
PCU-COM	19	6	

Name	AC 410 Execution Time, typical (µs)	AC 450, Execution Time, typical (µs)	Notes
PCU-I	100	34	
PCU-O	65	22	
PCU-SS	34	11	
PDP	147 31	50 10	BAL=0 BAL=1
PI	188	63	
PIDCON	MAN AUTO E1 BAL 805 820 - - 830 865 850 720 1050 1075 1085 940 1090 1140 1110 960 1110 1140 1150 1000 1090 1140 - -	MAN AUTO E1 BAL 270 275 - - 275 290 280 240 350 360 360 315 360 380 370 320 370 380 380 330 360 380 - -	 BAS BAS+(C1...C3) BAS+(C1...C4) BAS+(C1...C5) BAS+(C1...C6) BAS+(C4...C5)
PIDCONA	Not valid for AC 410	MAN AUTO E1 BAL 404 396 – - 437 427 417 415 450 475 450 4 32 530 560 590 4 51 736 780 786 4 78 505 534 – –	 BAS BAS+(C1...C3) BAS+(C1...C4) BAS+(C1...C5) BAS+(C1...C6) BAS+(C4...C5)
PIP	151 29	50 10	BAL=0 BAL=1
POS-A	180 420	60 140	C1=0 C1=1
POS-L	195	65	
POS-O	380 450	125 150	C1=0 C1=1
PRINT	13 1200+120×C1	5 400+40×C1	not active active
PULSE-S	42	14	
RAMP	72	24	
RAMP-S1	198	66	
RATIOSTN	MAN AUTO E1 345 300 305 425 380 425 450 410 440	MAN AUTO E1 115 100 100 140 125 140 150 135 145	 BAS BAS+C1 BAS+C1+C2

Name	**AC 410 Execution Time, typical (μs)**	**AC 450, Execution Time, typical (μs)**	**Notes**
REG	22+3×C2 22+4×C2 20+5×C2 20+4×C2	7+1×C2 7+1.3×C2 7+1.7×C2 7+1.3×C2	C1=I, B C1=IL, R C1=T C1=TR
REG-G	18+2×C2+1×C3×C4 18+2×C2+1.4×C3×C4 15+2×C2+C3(10+0.8×C4) 20+2×C2+C3(3+C4)	6+0.7×C2+0.3×C3×C4 6+0.7×C2+0.5×C3×C4 5+0.7×C2+C3(3+0.25×C4)’ 7+0.7×C2+C3(1+0.3×C4)	C1=I, C1=IL, R C1=B C1=T, TR
REG-RET	69+4×C2	23+1.3×C2	
REPORT	7 3400	2.5 1100	not active active
SCALE	480	160	
SCALEDOS	225	75	
SENDREQ	430 35	145 12	active not active
SEQ	260	90	C3=C4=C5=0, including STEP
SHIFT	13	4.3	
SHIFT-L	25	8	
SLAVEM	80	25	
SQRT	73	25	
SR	6	2	
SR-D	10	3.3	
SR-AA SR-AO SR-OA SR-OO	6+3×q 12+3×q 11+3×q 10+3×q	2+1×q 4+1×q 4+1×q 3+1×q	q=number of tested inputs
STEP	The execution time is included in SEQ	The execution time is included in SEQ	
SUB	11 11 26	4 4 8.5	C1=I C1=IL C1=R
SW	12+4×C2	4+1.3×C2	
SW-C	12+4×C2	4+1.3×C2	

Name	AC 410 Execution Time, typical (µs)	AC 450, Execution Time, typical (µs)	Notes
TBL-R TBL-W	55+5×C3+2×C4+585×T 65+5×C3+2×C4+585×T	20+2×C3+0.7×C4+195×T 22+2×C3+0.7×C4+195×T	T=0 if the input terminals are unchanged T=1 if any input terminals have changed since last execution and when the element is executed the first time. **Note:** The execution time will increase if TBLNAME is used. (The increase may be large).
TBL-RG, TBL-WG	N×700	N×230	N = the number of searched values. For instance, one value of data type IL is to be read. If TBLPARAM points to the first value of data type IL then N=1. If TBLPARAM points to a value of another data type, and the fourth value found is IL, then N=4. **Note** The execution time will increase if TBLNAME is used. (The increase may be large).
TEXT	52×C2+6.5×C4 4+85×C2+2×C4 200×C2+8×C4–160 20+9×C2 20+242×C2+1×C4 25+160×C2+5×C4 20+1×C2+6×C3	20×C2+2×C4 30×C2+1×C4 67×C2+2×C4–50 7+3×C2 7+80×C2+0.3×C4 8+50×C2+1.5×C4 20+0.4×C2+2×C3	C1=I C1=IL C1=R C1=B C1=T C1=TR C1=A
THRESH-L	9+2×q	3+0.7×q	q=number of tested inputs
TIME	25	8	
TIMER	46+14×C1	15+5×C1	
TOFF	6	2	
TON	6	2	
TON-RET	14	5	
TRIGG	7	2	
VALVECON	355+17×C1+23×C2	120+6×C1+7×C2	
XOR	9	3	

(1) If Output Variables is in E3 mode, the mode of the Controlled Variables has no effect on the execution time.

(2) These figures are for a small algorithm. A large algorithm may cause a considerably longer execution time.

Memory Space Requirements

The following pages show the memory space requirements for PC elements and their output terminals. The memory requirements described in the table below, regarding parameters and connections to data base elements, are added to the memory space requirements for PC elements. The memory space is allocated with the DIMPC command.

Type of Connection	**Data Type**	**Required Memory (byte)**	
		Local Data Area	**Total**
Operational Parameter or Constant(1)	I, B	2	6
	IL, R, T, TR	4	8
	A	2 + number of characters in Array	8+ number of characters in Array
Input terminal connected to Data Base Element	I, B	2	30
	IL, R, T, TR	4	32
	A	2 + number of characters in Array	32+ number of characters in Array
Connection to Element Output of a Different Priority	I, B	2	12
	IL, R, T, TR	4	14
	A	1 + number of characters in Array	14+ number of characters in Array
Output terminal connected to Data Base Element (2)	I, B, IL, R, T, TR, A	0	24
Variable assigned a Unit	-	0	6

(1) Multiple use of operational parameters and constants do not require any additional space.
(2) Additional connections to the same data base connections within a PC program requires 4 bytes per connection.

Name	PC statement (byte)	Local data area (byte)	Notes
ABS	18 18	2 4	C1=I C1=IL, R
ADD	14+2×C2 14+2×C2	2 4	C1=I C1=IL, R
ADD-MR	14 + 2 × (C1+C2)	4	
ADD-MR1	16+4 × C1	4	
ANALYSE	28+16 × C1+4 × C2 × (C1+1)	26+6 × C1+2 × C2	
AND	14+2 × C1	2	
AND-O	18+2×(C1+C2+C3+C4+C5+C6)	2	
BLOCK	28	2	
COM-AIS	48	42	
COM-CVI1	68	178	
COM-CVO1	50 84	70 134	C1=0 C1=1
COM-DIS	48	42	
COM-M	60	102	
COM-MP51	34+2×(C1+3×C2+(C2+1) × C3)	22+2 × C2	
COM-STAT	18+22×C1	2+26×C1	
COMP	30	12	
COMP-I	30	6	
COMP-R	26+8×(C1+C2)	8+2×(C1+C2)	
CON-PU1	72 96	90 90	C1=0 C1=1
CONTRM	68	6	
CONV	22 22	4 6	C2=I C2=IL, R, T, TR
CONV-AI	20	4	
CONV-BI	36+2×C3 36+2×C3	6 8	C1=I C1=IL
CONV-IA	32	16	
CONV-IB	42+2×C3	12+2×C3	
CONV-SA	46+8 × C2	8+C1	

Name	PC statement (byte)	Local data area (byte)	Notes
COUNT	50 50	10 12	C1=I C1=IL
COUNT-DP	38	12	
COUNT-L	66	14	
CVB-I	70+11 × (C1+C2)	112+8 × (C1+C2)	
CVB-O	32+4 × C1+9 × C2	284	
DATE	24	16	
DEMUX-MI	32+6 × C2 32+6 × C2	4+2 × C2 4+4 × C2	C1=I, B C1=IL, R, T, TR
DEMUXA-M	24+8×C2	2+(2+C1)×C2	
DER	54	44	
DISP	44	52	
DISP-SEG	38	52	
DIV	24 24	4 8	C1=I C1=IL, R
DIV-MR	14+2 × (C1+C2)	4	
DRI-CNV	22	4	
DRI-R	30 + 20 × C1	12 + 6 × C1	
DRI-S	24 + 8 × C1	16	
DP820-I	36	34	
DP820-O	22 + 12 x C1	10	
DSP-R	32 + 6 (C1+C2+C3+C4)	6 + 4 (C1+C3+C4) + 2×C2	
DSP-S	26 + 2 (C1+C2+C3+C4)	16	
EVENT	38+8 × C1+10 × C5	48+114 × C1	
EXP	22	6	
EXPAND	32+6 × C4	4+C1 × (C3+C4)	C1=2 if B or I, else 4
EXPAND-A	30	2+C2 2 4	C1=A C1=I C1=IL, R
FAULT	20+8 × C1	12+4 × C1	
FIFO	40+8 × C2 40+8 × C2	2 × (C2 + C2 × C3) 4 × (C2 + C2 × C3)	C1=I, B C1=IL, R, T, TR

Name	PC statement (byte)	Local data area (byte)	Notes
FIFO-RW	48+8 × C2+8 × C2 × C3 48+8 × C2+8 × C2 × C3	16+2 × (C2+C2 × C3) 16+4 × (C2+C2 × C3)	C1=I, B C1=IL, R, T, TR
FILT-1P	50	40	
FILT-2P	54	72	
FPM-COM	36+2×C1	22	
FPM-I	32+6×(C2+C4+C6+C8)	C1×C2+C3×C4+C5×C6+ C7×C8	C1, C3, C5, C7=2 if B or I, else 4
FPM-IA	32	22×C1	
FPM-O	28+2×(C2+C4+C6+C8)	20	
FPM-OA	28	20	
FREQ-MP	46	14	
FREQ-SP	70	26	
FUNCM	22	0	
FUNG-1V	42	36	
FUNG-2V	40	52	
FUNG-T	58	42	
FUZZYCON	66+2x(C1+C2)+36xC4+4xC5+ 6xC6+24xC8	156+14xC4+4xC5+ C6x(2+4xC7)+6xC8	
GENBIN-I	60	10	
GENBIN-O	80	20	
GENCON-I	58	10	
GENCON-O	50	14	
GENUSD-I	72	10	
GENUSD-O	110	32	
INT	52	28	
INV	14	2	
KEYB-FU	96	50	
KEYB-GR	32+14 × C1	140+4 × C1	
KEYB-N1	50+8 × C2+4 × C2 × C3	148+C2 × C3	C3=2 if B or I, else 4. Applies to the local data area.
LIM-N	52+6 × C2 52+6 × C2	14 20	C1=I C1=IL, R, T, TR

Name	PC statement (byte)	Local data area (byte)	Notes
LN	22	6	
LON-R	18+2xC2 + 2xC4 + 2xC2xC5 18+2xC2 + 2xC4 + 2xC2xC5 18+2xC2 + 2xC4 + 2xC2xC5	38 + C2xC3 38 + 2xC2 38 + 4xC2	C1 = A C1 = B, I C1 = IL, R, T
LON-S	18 + 2xC2	22	
MAJ-R	32+8 × C1	8+6 × C1	
MANSTN	112+16 × C1	94+8 × C1	
MASTER	70	6	
MAX	22+2 × C2 22+2 × C2	4 6	C1=I C1=IL, R
MED-R	14+2 × C1	4	
MFB-IN	28+6×(C1+C2+C3+C4+C5+C6)	4+4×(C1+C2+C3+C4+C5)+ C4×C6	
MFB-OUT	28+2×(C1+C2+C3+C4+C5+C6)	4	
MIN	22+2 × C2 22+2 × C2	4 6	C1=I C1=IL, R
MMC-IND	30+2 (C1+C2+C3+C4)	34	
MMC-ORD	60+6 × C1	28+2× C1	
MONO	30	12	
MOTCON	128+20 × C1+6 × C2+14 × C3+22 × C4+6 × C5+26 × C6	192+4 × C1+2 × C3+4× C4+6 × C6	
MOVE	8+8 × C2 8+8 × C2	2 × C2 4 × C2	C1=B, I C1=IL, R, T, TR
MOVE-A	8+10 × C2	C2(2+C1)	
MUL	14+2 × C2 14+2 × C2	2 4	C1=I C1=IL, R
MUX-I	22+2 × C2 22+2 × C2	4 6	C1=B, I C1=IL, R, T, TR
MUX-MI	34+2 × C2 34+2 × C2	6 8	C1=B, I C1=IL, R, T, TR
MUX-MN	32+4 × C2 32+4 × C2	6 8	C1=B, I C1=IL, R, T, TR
MUX-N	20+4 × C2 20+4 × C2	4 6	C1=B, I C1=IL, R, T, TR

Name	PC statement (byte)	Local data area (byte)	Notes
MUXA-I	14+2 × C1	4+(C1+1) × (C2+2)	
MUXGE-MI	22 22	6 8	C1=B, I C1=IL, R, T, TR
MUXGR-MI	38+2 × C2 38+2 × C2	6+2 × C2 6+4 × C2	C1=B, I C1=IL, R, T, TR
NOVATUNE	50+C2 × 18	1190+122 × C2+60 × C1	
NUM-IN	36	58	
OR	14+2 × C1	2	
OR-A	16+2 × (C1+C2+C3+C4+C5+C6)	2	
OSC-B	24	12	
OSC-SIN	32	22	
OSC-SQW	24+4×C1	10	
P-1	48	14	
PB-DIAG	32+6×C1	6+2×C1	
PB-R	32+6×C2	14+2×C2 14+4×C2	C1=B,I C1=IL,R
PB-S	26+2×C2	12	
P-DEADB	40	10	
PCPGM	44	2	
PCU-COM	16	22	
PCU-I	46	54	
PCU-O	38	26	
PCU-SS	32	26	
PDP	52	48	
PI	62	36	
PIDCON	246+16 × C1+32 × C2+8 × C5+4 × C6	358+8 × C1+16 × C4+12 × C5 +18 × C6	
PIDCONA	66+26×C1+12×C2+16×C3+18× C4+12×C5+16×C6	44+8xC1+16xC4+2xC5+4xC6 (for QC02-LIB22*7.0/0: 2162)	
PIP	52	44	
POS-A	102+90 × C1	308	

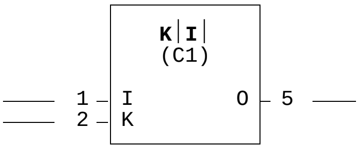
Name	PC statement (byte)	Local data area (byte)	Notes
POS-L	102	164	
POS-O	182+8 × C1	288	
PRINT	30+2 × C1	8	
PULSE-S	26	4	
RAMP	54	52	
RAMP-S1	52	54	
RATIOSTN	152+24 × C1+12 × C2	122+12 × C1+18 × C2	
REG	20+8 × C2 20+8 × C2	2+2 × C2 2+4 × C2	C1=B, I C1=IL, R, T, TR
REG-G	40+2 × (C2+C3) 40+2 × (C2+C3)	6+2 × C5 6+4 × C5	C1=B, I C1=IL, R, T, TR
REG-RET	22+2 × (C3+2 × C4+C5+C6+C7)+2 × C2 × (2+C4+C5+C6+C7)	4+C4+2 × (C4+C5+C6+C7) 4+C4+4 × (C4+C5+C6+C7)	C1=B, I C1=IL, R, T, TR
REPORT	36	44	
SCALE	170	68	
SCALEDOS	108	36	
SENDREQ	52+2 × (C2+C3)	24	
SEQ	82+18 × C3+22 × C4+82 × C5	30+2 × C3+8 × C4+50 × C5	
SHIFT-L	18+12 × C2 18+12 × C2	2+2 × C2 2+4 × C2	C1=B, I C1=IL, R, T, TR
SHIFT	30 30	2+2 × C2 2+4 × C2	C1=B, I C1=IL, R, T, TR
SLAVEM	38+2 × C2	0	
SQRT	22	6	
SR	16	2	
SR-D	26	4	
SR-AA SR-AO SR-OA SR-OO	14+2 × (C1+C2)	2	
STEP	40+16 × C1+4 × C2+4 × C3+2 × C4	2+6 × C1	

Name	PC statement (byte)	Local data area (byte)	Notes
SUB	18 18	2 4	C1=I C1=IL, R
SW	12+8 × C2 12+8 × C2	2 × C2 4 × C2	C1=B, I C1=IL, R, T, TR
SW-C	10+10 × C2 10+10 × C2	2 × C2 4 × C2	C1=B, I C1=IL, R, T, TR
TBL-R	46 + 2 × C2 + 10 × C3	42 + X1 + 4×C3 + C1×C3×C4	X1 = 2 for C1 = B, I X1 = 4 for C1 = T, TR, R, IL X1=22 for C1 = A
TBL-RG	50 + 2 × C7	54 + 4 ×(C1+C2+C4+C5)+ 2 ×(C3+C6) + 22 × C7	
TBL-W	44 + 2 × C2 +10 × C3	44 + 2 × C3	
TBL-WG	64 + 2 × C7	58	
TEXT	18+4 × C2	2+C4	
THRESH-L	14+2 × C2	2	
TIME	12	4	
TIMER	22+16 × C1	12+2 × C1	
TOFF	28	12	
TON	28	12	
TON-RET	30	12	
TRIGG	20	4	
VALVECON	144+12 × C1+20 × C2	110+2 × C1+4 × C2	
XOR	16	2	

Absolute Value ABS

Summary

ABS (**ABS**olute value) is used to form the absolute value of an integer or a real number. The absolute value can be multiplied with an optional value.



PC element ABS

Call ABS (C1)

Parameter	Significance	Permissible values
C1	Data type	I, IL, R

Terminal Description

No	Name	Type	Description
1	I	I*C1*	**I**nput. Input for number, the absolute value of which is to be obtained.
2	K	O*C1*	Input for multiplication factor K.
5	O	O*C1*	**O**utput. Output for the product of the value at input K and the absolute value of the value at the input I.

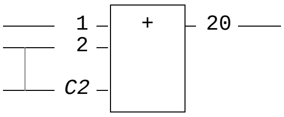
Function

The value at input K is multiplied by the absolute value of the input I and the result is stored at the output O.

Adder ADD

Summary

ADD is used for addition of an optional number of integers or real numbers



PC element ADD

Call ADD (C1, C2)

Parameter	Significance	Permissible values
C1	Data type	I, IL, R
C2	Number of inputs	2 - 19

Terminal Description

No	Name	Type	Description
1	-	I*C1*	Input for augend.
2	-	I*C1*	Input for addend.
: *C2*	-	I*C1*	Input for addend.
20	-	I*C1*	Output for sum.

Function

The values at the inputs 1 .. *C2* are added and the sum is stored at the output 20.

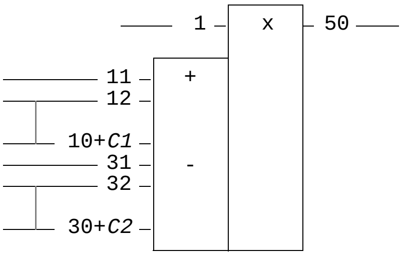
Overflow

If the maximum positive or negative values are exceeded, the output is limited to the highest or lowest representable value for the data type.

Adder with Multiplied Output ADD-MR

Summary

ADD-MR (**ADD**er **-** with **M**ultiplier **R**eal numbers) is used for addition of an optional number of real numbers. The numbers are added in two groups, each of a maximum of 19 numbers, after which the second group is subtracted from the first. The result is multiplied with a real number.



PC element ADD-MR

Call ADD-MR (C1, C2)

Parameter	Significance	Permissible values
C1	Number of positive inputs in the addition element.	1 - 19
C2	Number of negative inputs in the addition element.	1 - 19

Terminal Description

No	Name	Type	Description
1	-	IR	Input for multiplication factor.
11	-	IR	Input for augend with positive weight factor.
12	-	IR	Input for addend with positive weight factor.
: 10+*C1*	-	IR	Input for addend with positive weight factor.
31	-	IR	Input for addend with negative weight factor.
32	-	IR	Input for addend with negative weight factor.
: 30+*C2*	-	IR	Input for addend with negative weight factor.
50	-	OR	Output for result.

Function

The sum of the real numbers at the inputs 31 .. 30 + *C2* is subtracted from the sum of the inputs 11 .. 10 + *C1*, after which the result is multiplied with the value at input 1. The result is stored at the output 50.

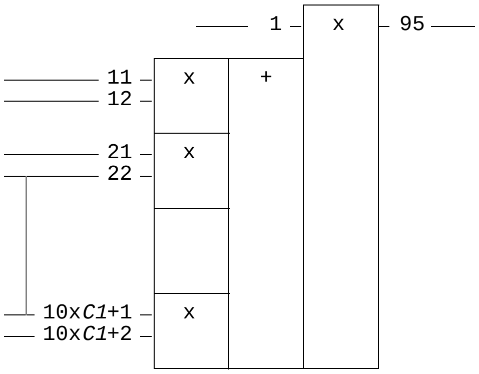
Overflow

If the maximum positive or negative real value is exceeded, the output is limited to the greatest or lowest representable value respectively.

Adder with Multiplied Output / Inputs ADD-MR1

Summary

ADD-MR1 (**ADD**er **-** with **M**ultiplier **R**eal numbers **1**) is used for addition of an optional number of products of real numbers. The result of the addition can then be multiplied with a real number.



PC element ADD-MR1

Call ADD-MR1 (C1)

Parameter	Significance	Permissible values
C1	Numbers of pairs of numbers to be added	2 - 9

Terminal Description

No	Name	Type	Description
1	--	IR	Input for multiplication factor.
11	--	IR	Input for augend.
12	--	IR	Input for weight factor.
21	--	IR	Input for addend.
22	--	IR	Input for weight factor.
: 10x*C1*+1	--	IR	Input for addend.
10x*C1*+1	--	IR	Input for weight factor.
95	--	OR	Output for result.

Function

The real values at the inputs 11, 21 .. 10x*C1*+1 are multiplied in pairs with the values at 12, 22 .. 10x*C1* + 2 respectively. The products are added and the sum is multiplied with the value at input 1. The result is stored at output 95.

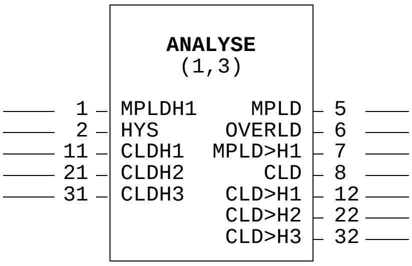
Overflow

If the maximum positive or negative real value is exceeded, the output is limited to the greatest or least representable value respectively.

System Load Measurement ANALYSE

Summary

ANALYSE is used for continuous measurement of the system load. It can also supervise the load from the control function part.



PC element ANALYSE

Call ANALYSE (C1, C2)

Parameter	Significance	Permissible values
C1	Selection of inputs MPLDH1 and HYS and outputs MPLD>H1 and CLD.	0 - 1
C2	Indicates the number of alarm levels (CLDHx) when supervising the control function load (if C1=1)	0 - 3

Terminal Description

No	Name	Type	Description
1	MPLDH1	IR	**M**ain **P**rocessor **L**oa**D** **H**igh limit **1**. Input for upper load limit. The value is given in percent. If the value is less than 0, the value 0 is used. If the value is greater than 100, the value 100 is used.
2	HYS	IR	**HYS**teresis. The hysteresis is given as the difference between the MPLD value at which the output MPLD>H1 is set to 1, and the value at which it is set to 0. If HYS is less than 0 the value 0 is used. If the value is greater than MPLDH1, the value MPLDH1 is used
5	MPLD	OR	**M**ain **P**rocessor **L**oa**D**. Output showing the total system load in percent.
6	OVERLD	OB	**OVERL**oa**D**. Output set to 1 when the system is overloaded.
7	MPLDH>H1	OB	**M**ain **P**rocessor **L**oa**D>H1**. Output set to 1 when the total load is greater than MPLDH1.
8	CLD	OR	**C**ontrol function **L**oa**D**. Output showing the load of the control function part. This value may not always be precise.
11	CLDH1	IR	**CLD** limit **H1**. Limit value 1 for the control function part in units of percent. If this value is less than 0, 0 is used. If the value is greater than 100, 100 is used.
12	CLD>H1	OB.	**C**ontrol function **L**oa**D>H1**. Output set to 1 when the control function load is greater than CLDH1.
13	CLDH2	IR	**CLD** limit **H2**. Limit value 2 for the control function part in units of percent. If this value is less than 0, 0 is used. If the value is greater than 100, 100 is used.
14	CLD>H2	OB	**C**ontrol function **L**oa**D>H2**. Output set to 1 when the control function load is greater than CLDH2.

Terminal Description

No	Name	Type	Description
:			
10+*C2*	CLDH*C2*	IR	**CLD** limit **H***C2*. Limit value *C2* for the control function part in units of percent. If this value is less than 0, 0 is used. If the value is greater than 100, 100 is used.
11+*C2*	CLD>H*C2*	OB	**C**ontrol function **L**oa**D>H***C2*. Output set to 1 when the control function load is greater than CLDH*C2*.

Function

The output OVERLD is set to 1 when the system is overloaded. The total system load is given in percent (%) at the output MPLD. This output is updated once per second.

If the call parameter *C1* is set to 1, the inputs MPLDH1 and HYS, and the outputs MPLD>H1 and CLD, are included.

The output MPLD>H1 is set to 1 when the load exceeds MPLDH1.

MPLD>H1 is cleared when the load falls below MPLDH1-HYS.

The value at the output CLD is stepped up to the value corresponding to the control function's load on the system, see the figure below.

The value at the output is in percent (%).

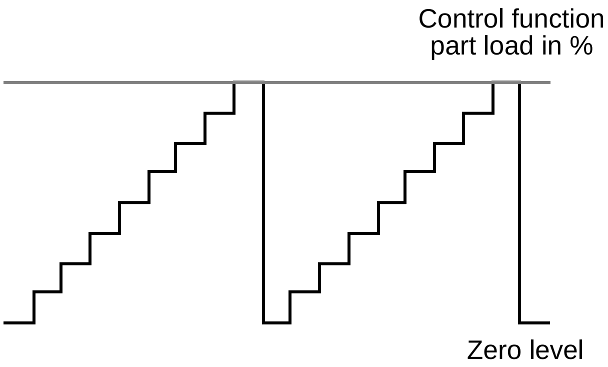
The number of limit values for supervision of CLD is chosen with call parameter *C2*.

The value at the output CLD is compared to limit values given at the inputs CLDH1 .. CLDC2.

When CLD exceeds a limit value the corresponding output CLD>H is set to 1.

ANALYSE should be executed with a cyclicity of at most 50 ms to obtain the best possible resolution at the output CLD.

.

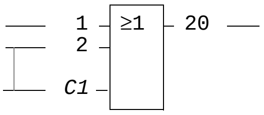


Function of the output CLD

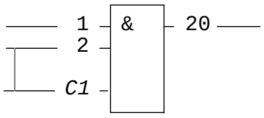
And Gate / Or Gate AND/OR

Summary

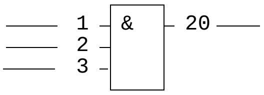
OR and AND elements are used to form general combinatory expressions with Boolean variables. The elements can have a maximum of 19 inputs.



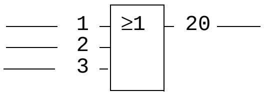
PC element OR



PC element AND



PC element AND (3)



PC element OR (3)

Call OR (C1) AND (C1)

Parameter	Significance	Permissible values
C1	Number of inputs	2 - 19

Terminal Description

No	Name	Type	Description
1	--	IB	Input
2	--	IB	Input
: *C1*	--	IB	Input
20	--	OB	Output

Function- OR element

The output signal is set if any of the inputs are set. See the table below.

Table OR

1	2	3	20
0	0	0	0
0	0	1	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

AND element

The output signal is set to 1 if all the inputs are 1. See the table below.

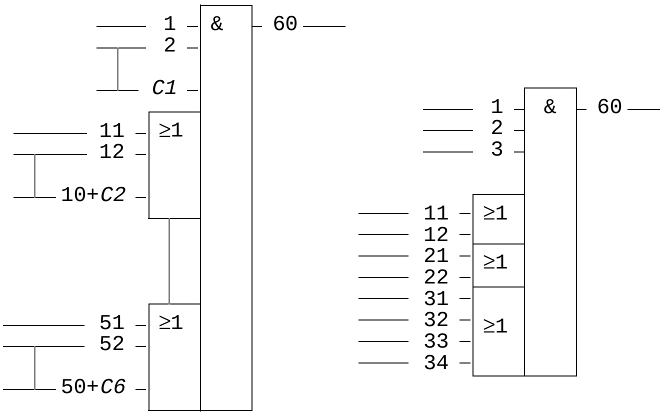
Table AND

1	**2**	**3**	**20**
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

And Gate With Or Inputs AND-O

Summary

AND-O (**AND -** with **O**r gates on the inputs) is used to form general combinatory expressions with Boolean variables.



PC element AND-O

Call AND-O (C1, C2, C3…C*n*)

Parameter	Significance	Permissible values
C1	Number of inputs in the AND part of the element	0 - 9
C2	Number of inputs in the first OR gate of the element	0 - 9
C3	Number of inputs in the second OR gate of the element	0 - 9
: C*n*(1)	Number of inputs in the last OR gate of the element	0 - 9

(1) $1 \leq n \leq 6$
$1 \leq C1+C2+...+Cn \leq 40$

Terminal Description

No	Name	Type	Description
1	--	IB	Input direct to the AND gate in the element.
2	--	IB	Input direct to the AND gate in the element.
: *C1*	--	IB	Input direct to the AND gate in the element.
11	--	IB	Input to the first OR gate in the element.
12	--	IB	Input to the first OR gate in the element.
: 10+*C2*	--	IB	Input to the first OR gate in the element.
21	--	IB	Input to the second OR gate in the element.
22	--	IB	Input to the second OR gate in the element.

Terminal Description

No	Name	Type	Description
: 20+*C3*	--	IB	Input to the second OR gate in the element.
: 10x(*n*-1)+1	--	IB	Input to the last OR gate in the element.
10x(*n*-1)+2	--	IB	Input to the last OR gate in the element.
: 10x(*n*-1)+C*n*	--	IB	Input to the last OR gate in the element.
60	--	IB	Output

Function

More rapid execution than if separate elements are used can be obtained by using assembled elements such as AND-O. The inputs need then only be tested until the value of the output can be determined. If input 1 is 0, the output can be reset irrespective of the other inputs. See the table below.

Table AND-O (3, 3, 3)

1	2	3	11	12	13	21	22	23	60
0	x(1)	x	x	x	x	x	x	x	0
1	0	x	x	x	x	x	x	x	0
1	0	0	x	x	x	x	x	x	0
1	1	1	0	0	0	x	x	x	0
1	1	1	0	0	1	0	0	0	0
1	1	1	0	0	1	0	0	1	1
1	1	1	0	0	1	0	1	x	1
1	1	1	0	0	1	1	x	x	1
1	1	1	0	1	x	0	0	0	0
1	1	1	0	1	x	0	0	1	1
1	1	1	0	1	x	0	1	x	1
1	1	1	0	1	x	1	x	x	1
1	1	1	1	x	x	0	0	0	0
1	1	1	1	x	x	0	0	1	1
1	1	1	1	x	x	0	1	x	1
1	1	1	1	x	x	1	x	x	1

(1) x indicates that the input has no effect on the value of the output or the execution time.

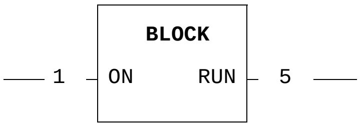
Block Header

BLOCK

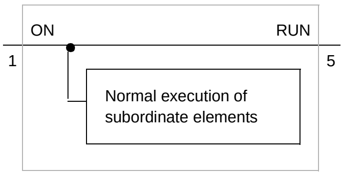
Summary

Block header BLOCK is used to connect and disconnect the execution of several PC elements.

Call BLOCK



PC element BLOCK



Function diagram

Terminal Description

No	Name	Type	Description
1	ON	IB	Control input which is to be set (to 1) for normal execution
5	RUN	OB	**RUN** indicates execution of the elements in the block.

Function

BLOCK is used to permit the connection and disconnection of a number of PC elements in a control module, slave or sequence step. A block may not contain structure elements.

Execution

For normal block execution, the superior execution unit must be under normal execution, and input ON of the block header must be set to 1. The block execution is then performed according to the conditions of the superior execution unit. If input ON is set to 0 after the block is executed, the calculated data remains until the next time the block is executed. If the superior execution unit is executed in the reset mode, the elements in the block will be executed in the same mode, irrespective of the state of input ON.

Blocking

Blocking requires input ON to be set to 0. At reset, the elements in the block will stop executing, but the latest value at each output will be read out to the data base unchanged. To reset the outputs, the superior execution unit must be executed in the reset mode. The superior execution unit will then read out 0-values to the data base. Blocking from a service aid is performed independently of the ON input of the block in accordance with the status of the execution unit which includes the block.

NOTE

time delays under BLOCK, for example, TOFF, will increase the delay time with the time of blocking.

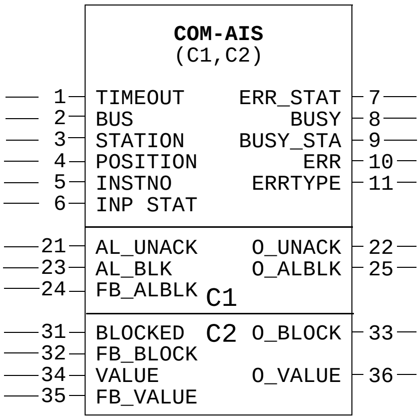
RUN

The output RUN is set only if normal execution is in progress, that is, the superior execution unit is under normal execution and the ON input of the block is set.

Send Commands to AIS in AC 100 Series COM-AIS

Summary

COM-AIS is used for sending commands to DB element AIS objects in Advant Controller 100 Series on Advant Fieldbus 100. Functions of the element are alarm acknowledge, alarm blocking, block update and value setting.



PC element COM-AIS

Call COM-AIS (C1,C2)

Parameter	Significance	Permissible values
C1	Enabling of alarm handling. If C1 is set to 1, the alarm acknowledge and alarm block functions are enabled.	0 - 1
C2	Enabling of value setting. If C2 is set to 1, the blocking and value setting functions are enabled.	0 - 1

Terminal Description

No	Name	Type	Description
1	TIMEOUT	IT	**TIMEOUT** value (in seconds) for command handling. Denotes the maximum wait time for the AC 100 Series controller to give response
2	BUS	II	**BUS** part of address.
3	STATION	II	**STATION** number of the AC 100 Series controller.
4	POSITION	II	**POSITION** of the AC 100 Series controller CPU.
5	INSTNO	II	**INST**ance **NO** of the AIS object to communicate with.
6	INP_STAT	IB	**INP**ut **STAT**us. This terminal should be connected to the error status bit in the DAT(B) holding the status word from the AIS object in the AC 100 Series controller.
7	ERR_STAT	OB	**ERR**or **STAT**us which is taken from the INP_STAT input.This terminal must be connected to the ERR property of AIEV
8	BUSY	OB	**BUSY** flag. Set to 1 if command pending, else 0.

Terminal Description

No	Name	Type	Description
9	BUSY_STA	OI	**BUSY STA**tus, may take one of theses values: 0 : no command pending 1 : update block command pending 2 : set value command pending 4 : alarm block command pending 5 : alarm acknowledge command pending
10	ERR	OB	**ERR**or. Output which is set to 1 if error exists when executing the element.
11	ERRTYPE	OI	Context related **ERR**or **TYPE**. Refer to the “Error Codes” section for error code details.
21	AL_UNACK	IB	**AL**arm **UNACK**nowledge. Acknowledge command is sent when this terminal changes from 1 to 0. Must be connected to the AL_UNACK terminal of the corresponding AIEV element. Terminal exists only if C1 is set to 1.
22	O_UNACK	OB	**O**utput alarm **UNACK**nowledge. Must be connected to the AL_UNACK terminal of the corresponding AIEV element. This terminal is updated from the AL_UNACK input terminal in the case of successful command execution. Whenever an error (including time-out) is encountered, the value on this terminal is set to the last value that caused a successful command completion. Terminal exists only if C1 is set to 1.
23	AL_BLK	IB	**AL**arm **BL**oc**K**. Alarm block command is sent when this terminal changes value. Must be connected to the AL_BLK terminal of the corresponding AIEV DB element. Terminal exists only if C1 is set to 1.
24	FB_ALBLK	IB	**F**eed**B**ack **AL**arm **BL**oc**K**. Must be connected to the alarm block bit in the DAT(B) holding the AIS status word. Terminal exists only if C1 is set to 1.
25	O_ALBLK	OB	**O**utput **AL**arm **BL**oc**K**. Must be connected to the AL_BLK terminal of the corresponding AIEV element. This terminal is updated from the AL_BLK input terminal in the case of successful command execution. Whenever an error (including time-out) is encountered or when FB_ALBLK changes, the value on this terminal is updated from the feedback value FB_ALBLK. Terminal exists only if C1 is set to 1.
31	BLOCKED	IB	**BLOCKED**. Block command is sent when the terminal changes value. Must be connected to the BLOCKED terminal of the corresponding AIEV element. Exists only if C2 is set to 1.
32	FB_BLOCK	IB	**F**eed**B**ack **BLOCK**. Must be connected to the block bit in the DAT(B) holding the AIS status word. Terminal exists only if C2 is set to 1.
33	O_BLOCK	OB	**O**utput **BLOCK**. Must be connected to the BLOCKED terminal of the corresponding AIEV element. This terminal is updated from the BLOCKED input terminal in the case of successful command execution. Whenever an error (including time-out) is encountered or when FB_BLOCK changes, the value on this terminal is updated from the feedback value FB_BLOCK. Terminal exists only if C2 is set to 1.
34	VALUE	IR	**VALUE**. Set value command is sent when the terminal changes value. Must be connected to the VALUE terminal of the corresponding AIEV element. Terminal exists only if C2 is set to 1.
35	FB_VALUE	IR	**F**eed**B**ack **VALUE**. Must be connected to a DAT(R) element receiving the feedback signal from the AC 100 Series controller AIS object. Terminal exists only if C2 is set to 1.

Terminal Description

No	Name	Type	Description
36	O_VALUE	OR	**O**utput **VALUE**. Must be connected to the VALUE terminal of the corresponding AIEV element. This terminal is updated from the VALUE input terminal in the case of successful command execution. Whenever an error (including time-out) is encountered or when FB_VALUE changes, the value on this terminal is updated from the feedback value FB_VALUE. Terminal exists only if C2 is set to 1.

Function, General

COM-AIS is used for sending commands towards AIS objects in the Advant Controller 100 Series over Advant Fieldbus 100. There are two command groups, one for Alarm Handling (Alarm Acknowledge and Alarm Blocking) and one for Value Settings (Update Block and Value Setting). Alarm Handling is enabled by the C1 call parameter, Value Setting is enabled by the C2 call parameter.

COM-AIS provides a complete set of terminals to identify a single Advant Controller 100 Series AIS object (BUS, STATION, POSITION, INSTNO), and terminals for detecting changes on the AL_UNACK, AL_BLK, BLOCKED and VALUE terminals of the connected DB element AIEV.

When an AIEV connected terminal changes state, corresponding output terminal will be updated immediately and a command will be issued towards the remote object (for alarm acknowledge a command is sent only if the AL_UNACK changes from 1 to 0). BUSY is raised and BUSY_STATUS is set according to the pending command type. At command completion, the BUSY and BUSY_STATUS are both cleared.

If, for some reason, the remote object did not receive the command or by any reason failed when executing the command, the corresponding AIEV connected output will be reset. For the Alarm Block, Update Block, and Value Setting commands this is accomplished by setting the corresponding output to the feedback value received from the remote object. For the Alarm Acknowledge command, the last successful command value will be set on the O_UNACK output.

Once setting a value on the AIEV connected outputs, the corresponding inputs will take the same value because of the tight configuration between COM-AIS and the AIEV element.

Only one command may be launched each time the element is executed. Alarm Acknowledge has the highest priority, followed by Alarm Block, Update Block and Value Setting (in this order).

Data is received from Advant Controller 100 Series with DB element DSP (with the related DAT(B)/DAT(R) elements).

Configuration

The communication is configured by means of:

- Pairs of COM-AIS/AIEV elements for each remote object to communicate with.
- Preconnected COM-AIS terminals as stated in the “Terminal Description” section.
- Running DSP/DAT elements on both sides for the remote objects to communicate with.

Error Codes

The outputs ERR and ERRTYPE specify the error states that can arise. ERR is a logic output that is set to 1 when an error is detected in the transmission of data from the node. ERRTYPE mirrors the error state of the element. The table below shows the various types of error that are mirrored in ERRTYPE.

Error Types

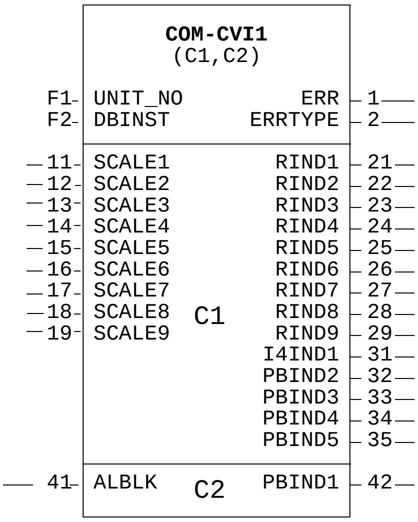
Error Code[1]	**Description of Error**
0	OK
1	Internal queue (channel) is full. The AC450 is probably overloaded. Reduced load in the system.
2	Bad signal channel index. Internal error. Report to ABB.
5	Command time-out. The PC element has given a command to AC 100 Series, but the Feedback terminal value has not been updated. Verify the connection of the feedback terminal; that the DSP is valid (valid = 1), that the AC 100 Series controller is operating satisfactory.
7	Physical bus number not found. The terminal BUS has a value of a AF100 bus number that is not properly configured in the AC 450. Check the AF 100 data base element, and that the DSP is valid (valid = 1).
8	Bad signal type. Internal error. Report to ABB.
9	Internal memory error. Internal error. Report to ABB.
10240 - 10495	Communication failure. The AF 100 is probably overloaded.
10848 - 15103	Communication failure. The AF 100 is probably overloaded.
14850	Illegal STATION number.
14857	Wrong INSTNO number.
14859	Wrong POSITION number.
14873	Wrong STATION number.
Terminal ERR is set to 1 if ERRTYPE <> 0.	

(1) Other numbers could appear, meaning internal error.

Data from Convertor COM-CVI1

Summary

COM-CVI1 is used as a communication link between a PC program in a controller and a convertor, TYRAK L. The element is specially designed to fetch information from the convertor. For communication, a data base element of type CV unit is required.



PC element COM-CVI1

Call COM -CVI1(C1,C2)

Parameter	Significance	Permissible values
C1	Indicates if arithmetic variables are to be included in the element. (Connection 11-35).	0 - 1
C2	Indicates whether MMC connection is to be included in the element. (Connection 41 and 42).	0 - 1

Terminal Description

No	Name	Type	Description
F1	UNIT_NO	FI	Number of the unit. Permitted values are 100-115, 200-215, 300-315, 400-415, that is, 100 x Bus+Jumper address.
F2	DBINST	FI	Instance number of the associated data base element MMCX. If set to 0, no events are transmitted to MMC.
1	ERR	OB	Error on initializing or outputs not updated.
2	ERRTYPE	OI	Error code

Terminal Description

No	Name	Type	Description
C1 dependent terminals:			
11	SCALE1	IR	Scale factor for output 21.
12	SCALE2	IR	Scale factor for output 22.
13	SCALE3	IR	Scale factor for output 23.
14	SCALE4	IR	Scale factor for output 24.
15	SCALE5	IR	Scale factor for output 25.
16	SCALE6	IR	Scale factor for output 26.
17	SCALE7	IR	Scale factor for output 27.
18	SCALE8	IR	Scale factor for output 28.
19	SCALE9	IR	Scale factor for output 29.
21	RIND1	OR	REAL 1 from convertor.
22	RIND2	OR	REAL 2 from convertor.
23	RIND3	OR	REAL 3 from convertor.
24	RIND4	OR	REAL 4 from convertor.
25	RIND5	OR	REAL 5 from convertor.
26	RIND6	OR	REAL 6 from convertor.
27	RIND7	OR	REAL 7 from convertor.
28	RIND8	OR	REAL 8 from convertor.
29	RIND9	OR	REAL 9 from convertor.
31	I4IND1	OIL	I x 4 from convertor (NB! see ”Continuous data”).
32	PBIND2	OI	Block word for command acknowledgments.(1)
33	PBIND3	OI	Block word for status.(1)
34	PBIND4	OI	Block word for indication to MMC.(1)
35	PBIND5	OI	Block word for interlocking inputs etc.(1)
C2 dependent terminals:			
41	ALBLK	IB	Blocking of event analysis.
42	PBND1	OI	Block word for event handling.(1)

(1) PBIND1-5 each contain 16 logic signals. The content of these outputs is transparent to the element. The content can therefore be changed without changing the element.

Function, General

COM-CVI1 is used to transfer variables (indications of measured values) to a higher level ABB Master system from a digital convertor (TYRAK L). The Master Fieldbus is used for communication.

Arithmetical variables are transmitted on the bus as integers. The data type R (real) outputs are scaled against the relevant scale factor (base value) before the value goes to the output. The element is divided into functions that can be chosen to suit the application.

Scaling

In Master, arithmetical variables are usually represented as real. Since the node works internally with integers (and fractions), scaling must be done in Master. This is done in the element for real outputs. The scale factor is an ordinary input and can be connected to a property in the data base or specified as an operating parameter. The scale factor represents the highest positive number that can be represented as I x 2. Conversion from the integers of the node to a real value at the output is performed according to the following formula:

output (R)=node value (I x 2) x (scale factor (R) / 32768)

To reduce the load on the element, only the inputs that are actually used are scaled. The element checks if the scale factor is 0. If so, no scaling takes place and the corresponding output is set to 0. The quotient of the incoming scale factor and 32768 is calculated only every 20th cycle to reduce the load on the element. The result is buffered and used in other calculations.
The scale factor should therefore not be dynamic.

Several PC Elements per Node

Since the element only reads data, it is permissible for the element to occur in several calls. This is useful when the event analysis subfunction is required to be placed under a short cycle time, while still executing the remaining functions during the normal cycle time.

Continuous Data

The element accesses DSCS 131 without semaphoring. This means that only two bytes of continuous data can be guaranteed. This limitation improves performance.

Data base

The data base element CV unit is used to start and monitor communication. At startup, the element searches in the data base for the unit that matches the bus and node specified by the function parameter UNIT_NO.

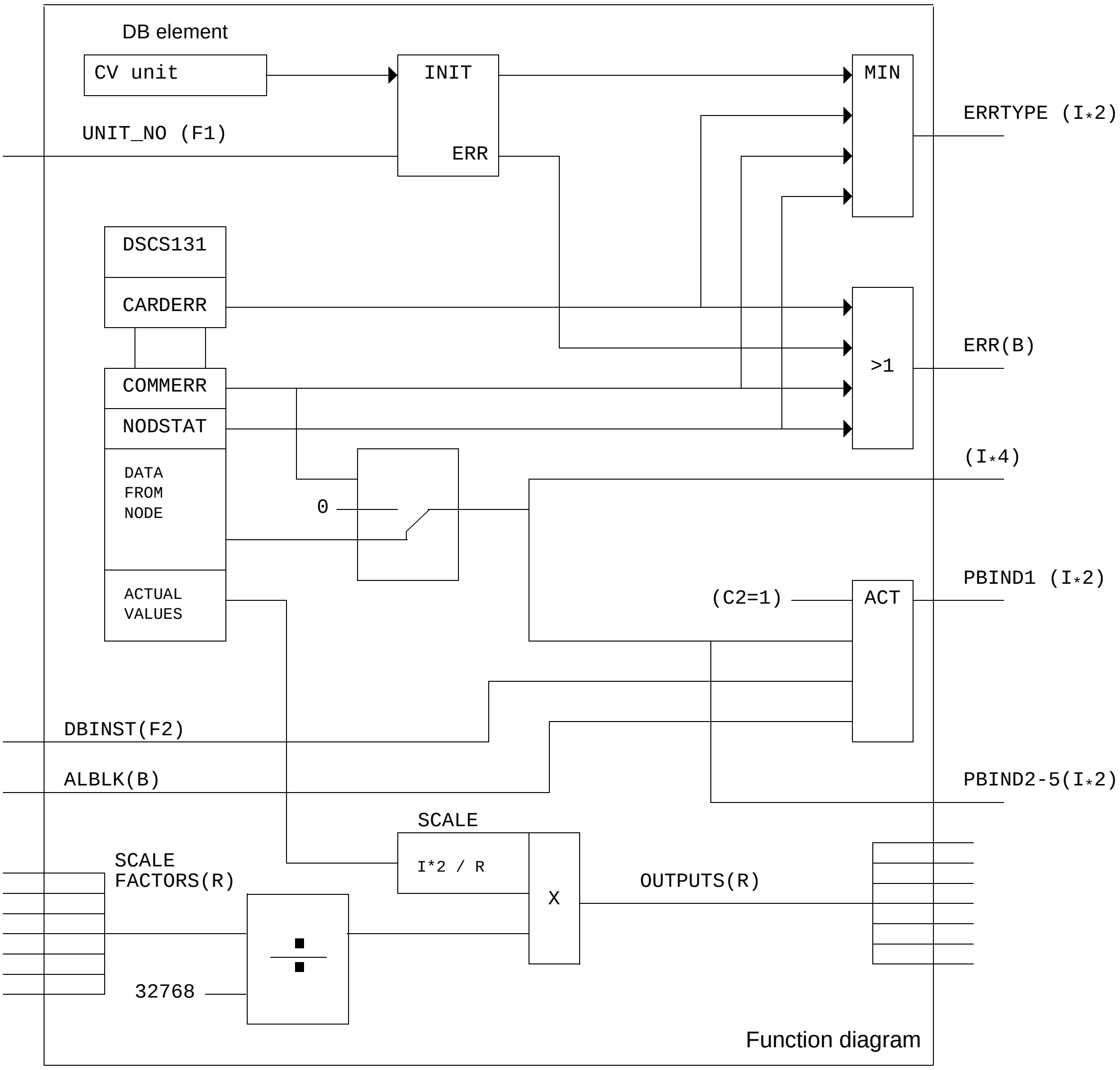
The same data base file is used for both order (COM-CVO1) and indication (COM-CVI1) elements.

Error Codes

The outputs ERR and ERRTYPE specify the error states that can arise. ERR is a logic output that is set to 1 when an error is detected in the transmission of data from the node. ERRTYPE mirrors the error state of the element but it continues to execute as usual. The table below shows the various types of error that are mirrored in ERRTYPE.

Error Types

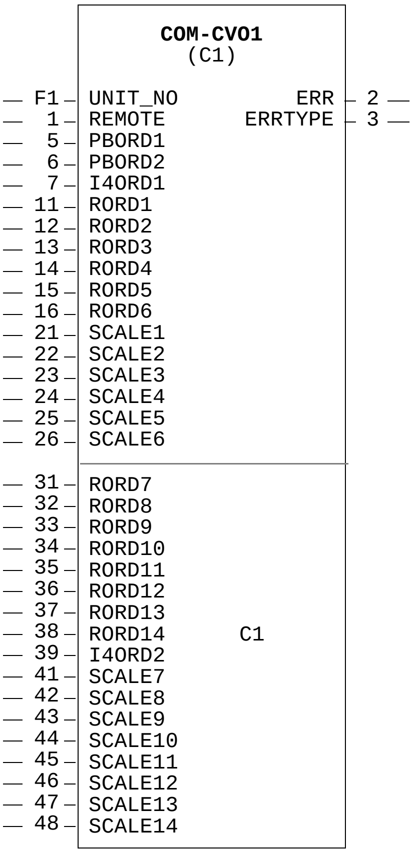
Error Code	Description of Error
–1	Communication with the convertor is not initiated.
–2	The convertor is not initiated from the controller.
00	OK
1	The data base element given by the PC element parameter UNIT_NO is missing.
2	The data base element given by the PC element parameter UNIT_NO is not of the CV unit type.
3	The data base element is not accepted. The reason may be that the parameter IMPL=0 or wrong bus number.
4	The bus control board DSCS 131 is missing or is not working correctly.
5	The convertor is missing on the bus. Communication error.
6	Initiating is taking place. Remaining indication indicates system error.
7	Connected node is not a convertor.


DB element
CV unit
INIT
ERR
MIN
UNIT_NO (F1)
ERRTYPE (I*2)
DSCS131
CARDERR
COMMERR
NODSTAT
DATA FROM NODE
ACTUAL VALUES
>1
ERR(B)
(I*4)
0
(C2=1)
ACT
PBIND1 (I*2)
DBINST(F2)
ALBLK(B)
PBIND2-5(I*2)
SCALE
I*2 / R
X
SCALE FACTORS(R)
OUTPUTS(R)
32768
Function diagram

Data to Convertor COM-CVO1

Summary

The COM-CVO1 element is used as a communication link between a PC program in a controller and a convertor, TYRAK L. The element is specially designed to give set points to the convertor. For communication, a data base element of type CV unit is required.



PC element COM-CVO1

Call COM-CVO1 (C1)

Parameter	Significance	Permissible values
C1	Indicates whether the input signals 31-48 are to be included in the element or not.	0 - 1

Terminal Description

No	Name	Type	Description
F1	UNIT_NO	FI	Number of the unit. Permitted values are 100-115, 200-215, 300-315, 400-415, that is, 100 x Bus + Jumper address.
1	REMOTE	IB	Node ordered to remote mode.

Terminal Description

No	Name	Type	Description
2	ERR	OB	Error on initializing or communication error.
3	ERRTYPE	OI	Error code.
5	PBORD1	II	Block term 1(1).
6	PBORD2	II	Block term 2(1).
7	I4ORD1	IL	I x 4 to convertor.
11	RORD1	IR	REAL 1 to convertor.
12	RORD2	IR	REAL 2 to convertor.
13	RORD3	IR	REAL 3 to convertor.
14	RORD4	IR	REAL 4 to convertor.
15	RORD5	IR	REAL 5 to convertor.
16	RORD6	IR	REAL 6 to convertor.
21	SCALE1	IR	Scale factor for input 11.
22	SCALE2	IR	Scale factor for input 12.
23	SCALE3	IR	Scale factor for input 13.
24	SCALE4	IR	Scale factor for input 14.
25	SCALE5	IR	Scale factor for input 15.
26	SCALE6	IR	Scale factor for input 16.
If *C1*=1, the following option values will be transmitted (if *C1*=0, default values=0 are transmitted for these variables):			
31	RORD7	IR	REAL 7 to convertor.
32	RORD8	IR	REAL 8 to convertor.
33	RORD9	IR	REAL 9 to convertor.
34	RORD10	IR	REAL 10 to convertor.
35	RORD11	IR	REAL 11 to convertor.
36	RORD12	IR	REAL 12 to convertor.
37	RORD13	IR	REAL 13 to convertor.
38	RORD14	IR	REAL 14 to convertor.
39	I4ORD2	IIL	I x 4 to convertor.
41	SCALE7	IR	Scale factor for input 31.
42	SCALE8	IR	Scale factor for input 32.
43	SCALE9	IR	Scale factor for input 33.
44	SCALE10	IR	Scale factor for input 34.

Terminal Description

No	Name	Type	Description
45	SCALE11	IR	Scale factor for input 35.
46	SCALE12	IR	Scale factor for input 36.
47	SCALE13	IR	Scale factor for input 37.
48	SCALE14	IR	Scale factor for input 38.

(1) PBORD1-2 each contain 16 logical variables. The content is transparent to the element except bit 15 in PBORD1, which is reserved for the REMOTE input of the element. The meaning can therefore be changed without changing the element.

Function, General

PC element COM-CVO1 is used to transfer variables (orders) from a higher level system to a digital convertor (TYRAK_ L). The Master Fieldbus is used for communication. The arithmetical input signals (R) of the element are rescaled and transmitted on the bus as integers (I x 2).

Scaling

The inputs of the element that are of the data type real (R) are rescaled to integers using a scale factor that is also of the data type real. This is done because integers are used internally in the node (TYRAK). The scale factor is an ordinary input and can be connected to a property in the data base or specified as an operating parameter. The scale factor represents the highest positive number that can be represented as I x 2. Conversion from real input value to an integer is performed according to the following formula:

output (I x 2)=input (R) x 32768 / scale factor (R)

To reduce the load on the element, only those inputs that are actually used are scaled. The element checks whether the scale factor is 0 and if it is, no scaling takes place and the default value of the node is transmitted (=0). The quotient of 32768 and the incoming scale factor is calculated only every 20th cycle to reduce the load on the element. The result is buffered and used in other calculations. The scale factor should therefore not be dynamic. The element limits data to the node between an upper limit (32767) and a lower limit (–32768). The hexadecimal equivalent of the maximum value 32767 is $7FFF and of the minimum value -32768 it is $8000.

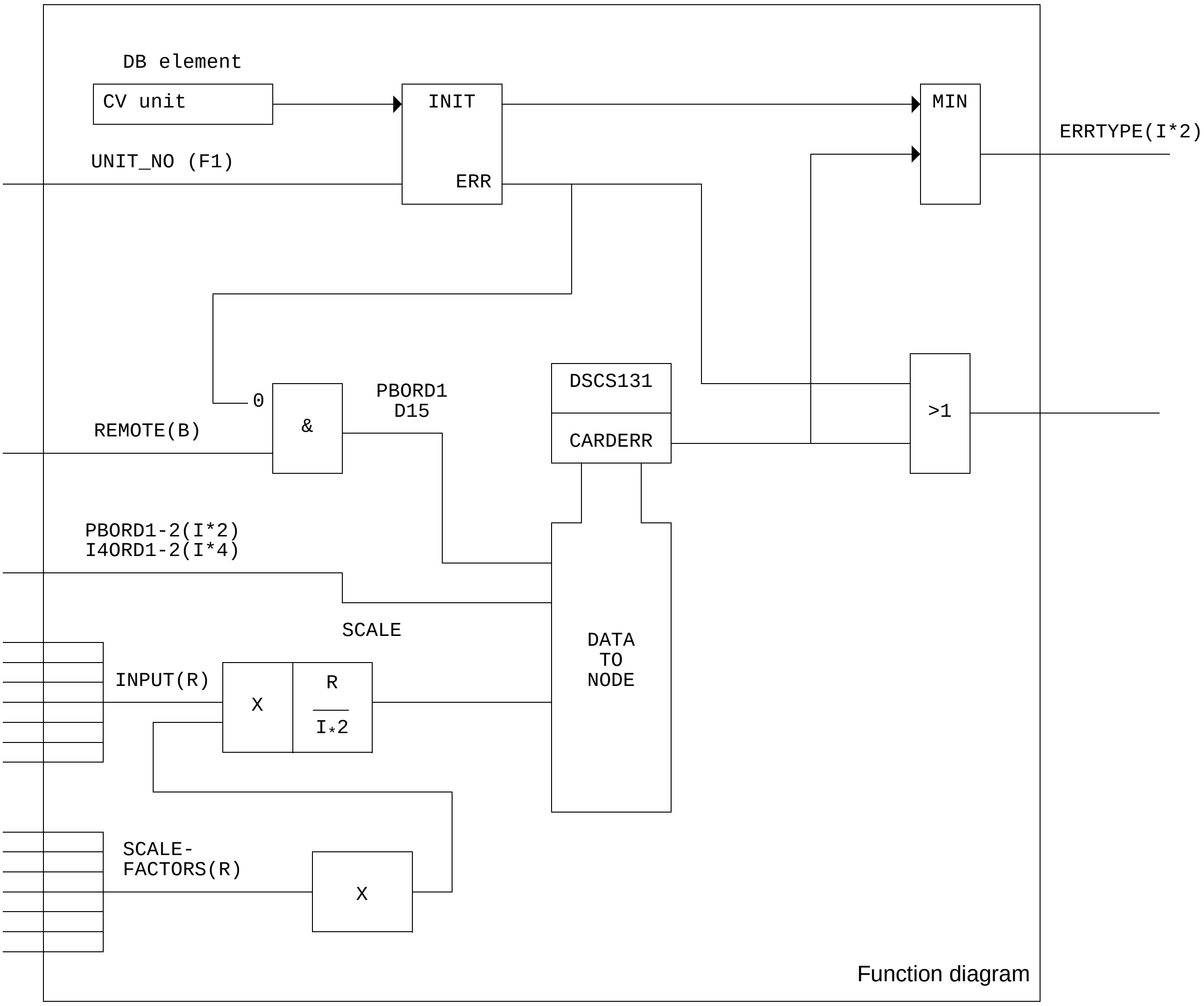
Data Base

The data base element CV unit is used to start and monitor communication. On starting, the element searches the data base for the unit that matches the bus and node specified by the function parameter UNIT_NO.

Error Codes

The outputs ERR and ERRTYPE specify the error states that can arise. ERR is a logic output that is set to 1 when an error is detected in the transmission of data to the node. ERRTYPE mirrors the error state of the element but it continues to execute as usual. The table shows the various types of error that are mirrored in ERRTYPE.

Error Code	Description of error
–1	Communication with the convertor is not initiated.
–2	The convertor is not initiated from the controller.
0	OK
1	The data base element given by the PC element parameter UNIT_NO is missing.
2	The data base element given by the PC element parameter UNIT_NO is not of the CV unit type.
3	The data base element is not accepted. The reason may be that the parameter IMPL=0 or wrong bus number.
4	The bus control board DSCS 131 is missing or not working correctly

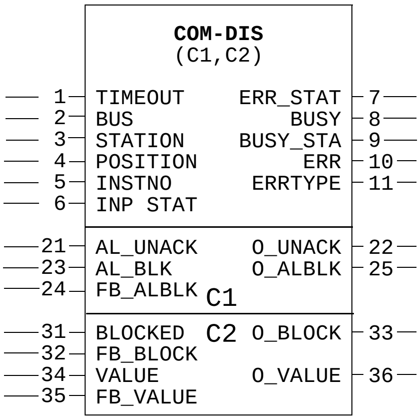


Function diagram

Send Commands to DIS in AC 100 Series COM-DIS

Summary

COM-DIS is used for sending commands to DB element DIS objects in Advant Controller 100 Series on Advant Fieldbus 100. Functions of the element are alarm acknowledge, alarm blocking, block update and value setting.



PC element COM-DIS

Call COM-DIS (C1,C2)

Parameter	Significance	Permissible values
C1	Enabling of alarm handling. If C1 is set to 1, the alarm acknowledge and alarm block functions are enabled.	0 - 1
C2	Enabling of value setting. If C2 is set to 1, the blocking and value setting functions are enabled.	0 - 1

Terminal Description

No	Name	Type	Description
1	TIMEOUT	IT	**TIMEOUT** value (in seconds) for command handling. Denotes the maximum wait time for the AC 100 Series controller to give response
2	BUS	II	**BUS** part of address.
3	STATION	II	**STATION** number of the AC 100 Series controller.
4	POSITION	II	**POSITION** of the AC 100 Series controller CPU.
5	INSTNO	II	**INST**ance **NO** of the DIS object to communicate with.
6	INP_STAT	IB	**INP**ut **STAT**us. This terminal should be connected to the error status bit in the DAT(B) holding the status word from the DIS object in the AC 100 Series controller.
7	ERR_STAT	OB	**ERR**or **STAT**us which is taken from the INP_STAT input.This terminal must be connected to the ERR property of DIEV
8	BUSY	OB	**BUSY** flag. Set to 1 if command pending, else 0.

Terminal Description

No	Name	Type	Description
9	BUSY_STA	OI	**BUSY STA**tus, may take one of theses values: 0 : no command pending 1 : update block command pending 2 : set value command pending 4 : alarm block command pending 5 : alarm acknowledge command pending
10	ERR	OB	**ERR**or. Output which is set to 1 if error exists when executing the element.
11	ERRTYPE	OI	Context related **ERR**or **TYPE**. Refer to the “Error Codes” section for error code details.
21	AL_UNACK	IB	**AL**arm **UNACK**nowledge. Acknowledge command is sent when this terminal changes from 1 to 0. Must be connected to the AL_UNACK terminal of the corresponding DIEV element. Terminal exists only if C1 is set to 1.
22	O_UNACK	OB	**O**utput alarm **UNACK**nowledge. Must be connected to the AL_UNACK terminal of the corresponding AIEV element. This terminal is updated from the AL_UNACK input terminal in the case of successful command execution. Whenever an error (including time-out) is encountered, the value on this terminal is set to the last value that caused a successful command completion. Terminal exists only if C1 is set to 1.
23	AL_BLK	IB	**AL**arm **BL**oc**K**. Alarm block command is sent when this terminal changes value. Must be connected to the AL_BLK terminal of the corresponding DIEV element. Terminal exists only if C1 is set to 1.
24	FB_ALBLK	IB	**F**eed**B**ack **AL**arm **BL**oc**K**. Must be connected to the alarm block bit in the DAT(B) holding the AIS status word. Terminal exists only if C1 is set to 1.
25	O_ALBLK	OB	**O**utput **AL**arm **BL**oc**K**. Must be connected to the AL_BLK terminal of the corresponding AIEV element. This terminal is updated from the AL_BLK input terminal in the case of successful command execution. Whenever an error (including time-out) is encountered or when FB_ALBLK changes, the value on this terminal is updated from the feedback value FB_ALBLK. Terminal exists only if C1 is set to 1.
31	BLOCKED	IB	**BLOCKED**. Block command is sent when the terminal changes value. Must be connected to the BLOCKED terminal of the corresponding DIEV element. Exists only if C2 is set to 1.
32	FB_BLOCK	IB	**F**eed**B**ack **BLOCK**. Must be connected to the block bit in the DAT(B) holding the DIS status word. Terminal exists only if C2 is set to 1.
33	O_BLOCK	OB	**O**utput **BLOCK**. Must be connected to the BLOCKED terminal of the corresponding DIEV element. This terminal is updated from the BLOCKED input terminal in the case of successful command execution. Whenever an error (including time-out) is encountered or when FB_BLOCK changes, the value on this terminal is updated from the feedback value FB_BLOCK. Terminal exists only if C2 is set to 1.
34	VALUE	IB	**VALUE**. Set value command is sent when the terminal changes value. Must be connected to the VALUE terminal of the corresponding DIEV element. Terminal exists only if C2 is set to 1.
35	FB_VALUE	IB	**F**eed**B**ack **VALUE**. Must be connected to a DAT(B) element receiving the feedback signal from the AC 100 Series DIS object. Terminal exists only if C2 is set to 1.
36	O_VALUE	OB	**O**utput **VALUE**. Must be connected to the VALUE terminal of the corresponding DIEV element. This terminal is updated from the VALUE input terminal in the case of successful command execution. Whenever an error (including time-out) is encountered or when FB_VALUE changes, the value on this terminal is updated from the feedback value FB_VALUE. Terminal exists only if C2 is set to 1.

Function, General

COM-DIS is used for sending commands towards DIS objects in the Advant Controller 100 Series over Advant Fieldbus 100. There are two command groups, one for Alarm Handling (Alarm Acknowledge and Alarm Blocking) and one for Value Settings (Update Block and Value Setting). Alarm Handling is enabled by the C1 call parameter, Value Setting is enabled by the C2 call parameter.

COM-DIS provides a complete set of terminals to identify a single Advant Controller 100 Series AIS object (BUS, STATION, POSITION, INSTNO), and terminals for detecting changes on the AL_UNACK, AL_BLK, BLOCKED and VALUE terminals of the connected DIEV object.

When a DIEV connected terminal changes state, corresponding output terminal will be updated immediately and a command will be issued towards the remote object (for alarm acknowledge a command is sent only if the AL_UNACK changes from 1 to 0). BUSY is raised and BUSY_STATUS is set according to the pending command type. At command completion, the BUSY and BUSY_STATUS are both cleared.

If, for some reason, the remote object did not receive the command or by any reason failed when executing the command, the corresponding DIEV connected output will be reset. For the Alarm Block, Update Block, and Value Setting commands this is accomplished by setting the corresponding output to the feedback value received from the remote object. For the Alarm Acknowledge command, the last successful command value will be set on the O_UNACK output.

Once setting a value on the DIEV connected outputs, the corresponding inputs will take the same value because of the tight configuration between COM-DIS and the DIEV element.

Only one command may be launched each time the element is executed. Alarm Acknowledge has the highest priority, followed by Alarm Block, Update Block and Value Setting (in this order).

Data is received from Advant Controller 100 Series with DB element DSP (with the related DAT(B) elements).

Configuration

The communication is configured by means of:

- Pairs of COM-DIS/DIEV elements for each remote object to communicate with.
- Preconnected COM-DIS terminals as stated in the “Terminal Description” section.
- Running DSP/DAT elements on both sides for the remote objects to communicate with.

Error Codes

The outputs ERR and ERRTYPE specify the error states that can arise. ERR is a logic output that is set to 1 when an error is detected in the transmission of data from the node. ERRTYPE mirrors the error state of the element. The table below shows the various types of error that are mirrored in ERRTYPE.

Error Types

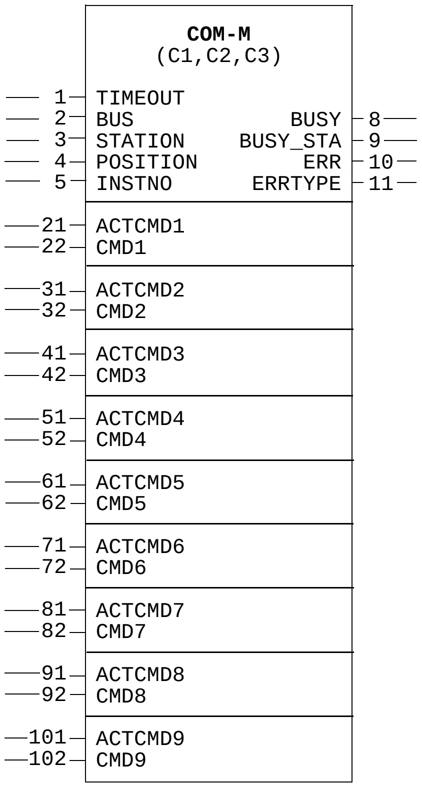
Error Code(1)	**Description of Error**
0	OK
1	Internal queue (channel) is full. The AC450 is probably overloaded. Reduced load in the system.
2	Bad signal channel index. Internal error. Report to ABB.
5	Command time-out. The PC element has given a command to AC 100 Series, but the Feedback terminal value has not been updated. Verify the connection of the feedback terminal; that the DSP is valid (valid = 1), that the AC 100 Series is operating satisfactory.
7	Physical bus number not found. The terminal BUS has a value of a AF100 bus number that is not properly configured in the AC 450. Check the AF 100 data base element, and that the DSP is valid (valid = 1).
8	Bad signal type. Internal error. Report to ABB.
9	Internal memory error. Internal error. Report to ABB.
10240 - 10495	Communication failure. The AF 100 is probably overloaded.
10848 - 15103	Communication failure. The AF 100 is probably overloaded.
14850	Illegal STATION number.
14857	Wrong INSTNO number.
14859	Wrong POSITION number.
14873	Wrong STATION number.
Terminal ERR is set to 1 if ERRTYPE <> 0.	

(1) Other numbers could appear, meaning internal error.

Send Commands to MB/MI/MIL/MR in AC 100 Series COM-M

Summary

COM-M (COMmands - MB/MI/MIL/MR) is used for sending commands of boolean, integer, long integer or real type values to MB/MI/MIL/MR objects in Advant Controller 100 Series on Advant Fieldbus 100.



PC element COM-M

Call COM-M (C1, C2, C3)

Parameter	Significance	Permissible values
C1	Defines the number of command groups.	1 - 9
C2	Defines the data type of the CMD*C1* terminals. C2=I, 16 bit integer value is sent. C2=R, 32 bit real value is sent. C2=IL, refer to C3.	I,IL,R
C3	Defines if command is to be sent to MIL or MB (only effective when C2=IL). C3=1, 32 bit long integer value is sent. C3=0, 32 bit long bit-pattern is sent.	0 - 1

Terminal Description

No	Name	Type	Description
1	TIMEOUT	IT	**TIMEOUT** value (in seconds) for command handling. Denotes the maximum wait time for the AC 100 Series controller to give response
2	BUS	II	**BUS** part of address.
3	STATION	II	**STATION** number of the AC 100 Series controller.
4	POSITION	II	**POSITION** of the AC 100 Series controller CPU.
5	INSTNO	II	**INST**ance **NO** of the MB/MI/MIL/MR object to communicate with.
8	BUSY	OB	**BUSY** flag. Set to 1 if command pending, else 0.
9	BUSY_STA	OI	**BUSY STA**tus, may take one of theses values: 0 : no command pending 1 : update block command pending 2 : set value command pending 4 : alarm block command pending 5 : alarm acknowledge command pending
10	ERR	OB	**ERR**or. Output which is set to 1 if error exists when executing the element.
11	ERRTYPE	OI	Context related **ERR**or **TYPE**. Refer to the “Error Codes” section for error code details.
	ACTCMD*C1*	IB	**ACT**ivate **C**o**M**man**D**. Whenever this terminal is raised high, the corresponding command value CMD is sent to the AC 100 Series controller object.
	CMD*C1*	I*C2*	**C**o**M**man**D**. This terminal holds the command value to be sent when corresponding ACTCMD terminal goes from 0 to 1.

Function

COM-M is used for sending commands towards MB/MI/MIL/MR objects in Advant Controller 100 Series over Advant Fieldbus 100. There are up to nine command groups (the count is defined by the C1 call parameter). The destination object in Advant Controller 100 Series is defined by the call parameter C2 (and C3 in case of MB or MIL).

COM-M provides a complete set of terminals to identify a single Advant Controller 100 Series MB/MI/MIL/MR object (BUS, STATION, POSITION, INSTNO), and terminals for defining command values and trigging the command sending.

When an ACTCMD terminal changes value from 0 to 1, the corresponding CMD value is sent to the VALUE property of the Advant Controller 100 Series corresponding object.

Only one command may be launched each time the element is executed. There exists a buffering of the ACTCMD values within the element. Command activations traced at different execution cycles is ordered by time (FIFO buffer). When several ACTCMD terminals are triggered within the same execution cycle, the command groups are prioritized by the call parameter C1 in increasing order (ACTCMD1 having the highest priority).

The command buffer are dimensioned to hold up to 10 commands. The buffer size is static (independent of the command group count). In a full buffer, new commands will be written into the last buffer position only.

Configuration

The communication is configured by means of:

- Preconnected COM-M terminals as stated in the “Terminal Description” section.

Error Codes

The outputs ERR and ERRTYPE specify the error states that can arise. ERR is a logic output that is set to 1 when an error is detected in the transmission of data from the node. ERRTYPE mirrors the error state of the element. The table below shows the various types of error that are mirrored in ERRTYPE.

Error Types

Error Code[1]	Description of Error
0	OK
1	Internal queue (channel) is full. The AC450 is probably overloaded. Reduced load in the system.
2	Bad signal channel index. Internal error. Report to ABB.
7	Physical bus number not found. The terminal BUS has a value of a AF100 bus number that is not properly configured in the AC 450. Check the AF 100 data base element, and that the DSP is valid (valid = 1).
8	Bad signal type. Internal error. Report to ABB.
9	Internal memory error. Internal error. Report to ABB.
10240 - 10495	Communication failure. The AF 100 is probably overloaded.
10848 - 15103	Communication failure. The AF 100 is probably overloaded.
14850	Illegal STATION number.
14857	Wrong INSTNO number.
14859	Wrong POSITION number.
14873	Wrong STATION number.
Terminal ERR is set to 1 if ERRTYPE <> 0.	

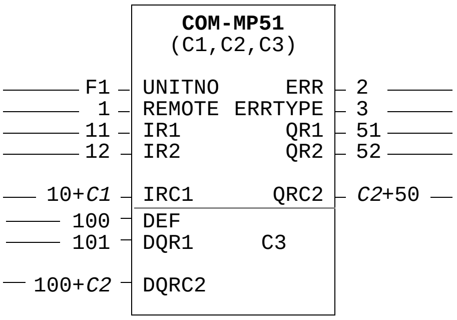
(1) Other numbers could appear, meaning internal error.

Communication with MP 51

COM-MP51

Summary

COM-MP51 (**COM**munication **- M**aster**P**iece **51**) is used for transmission of data between a controller and a MasterPiece 51 via the Master Fieldbus.



PC element COM-MP51

Call COM-MP51 (C1,C2,C3)

Parameter	Significance	Permissible values
C1	Number of data inputs	0 - 32
C2	Number of data outputs	0 - 32
C3	Selection of pre-defined values, that is, the possibility of replacing reading from the MP 51 by user-defined values at the outputs QR at fault or during testing. If C3 is not given it is set to 0.	0 - 1

Terminal Description

No	Name	Type	Description
F1	UNITNO	FI	**UNIT N**umber. Connection to the data base element. The hundreds indicate the number of the bus. Only values greater than or equal to 100 may be given.
1	REMOTE	IB	Input for activation of data transmission from the controller to MP 51. If REMOTE=0 data is only transmitted from MP 51 to AC 400. At the same time the flag F103 in MP 51 is set to indicate that the communication from the controller has ceased. Normally this input is set to D=1.
2	ERR	OB	**ERR**or. Output which is set to 1 if errors exist at initialization or updating. The QR outputs are not updated from MP 51.
3	ERRTYPE	OI	**ERR**or **TYPE**. Output for error type at initialization or reading, see the Error Codes table. 0 implies that no errors have occurred.
11	IR1	IB	**I**nput **R**emote **1**. Input for value to input IR1 on MP 51.
12	IR2	IB	**I**nput **R**emote **2**. Input for value to input IR2 on MP 51.
10+*C1*	IR*C1*	IB	**I**nput **R**emote ***C1***. Input for value to input IR*C1* on MP 51.
51	QR1	OB	Output for value from the output QR1 on MP 51

Terminal Description

No	Name	Type	Description
52	QR2	OB	Output for value from the output QR2 on MP 51.
: 50+*C2*	QR*C2*	OB	Output for value from the output QR*C2* on MP 51.
100(1)	DEF	IB	**DEF**ault. DEF=1 causes QR to be set to default values. With DEF=0 QR is updated with values from MP 51. If DEFault is used during operation, it is normally connected to the ERRor output.
101(1)	DQR1	IB	**D**efault **QR1**. Input for value of QR1 with DEFault.
102(1)	DQR2	IB	**D**efault **QR2**. Input for value of QR2 with DEFault.
: 100+*C2*(1)	DQR*C2*	IB	**D**efault **QR**italic=***C2***. Input for value of QR*C2* with DEFault.

(1) These terminals are only included if *C3*=1.

Function

A more detailed description of the function and use of the element is available in "MasterFieldbus and S400 I/O - User’s Guide". A data base element - PX - is needed for startup of MP 51. This is described in the data sheet for the data base element and in the Manual ”Data Base Elements”. Data transmission from the controller to MP 51 is activated by setting REMOTE=1. Data transmission from MP 51 to the controller is activated if DEF=0 or *C3*=0. A schematic view of COM-MP 51's function is given in the figure Function diagram on next page.

Fault Handling

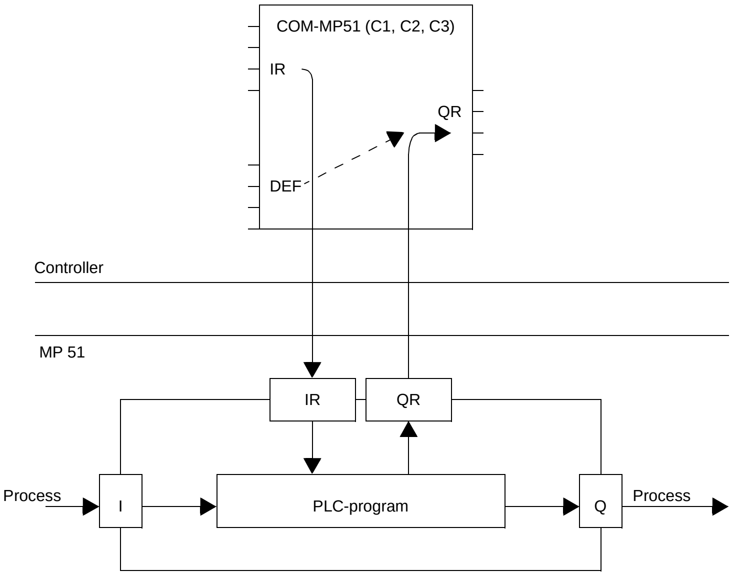
The element distinguishes between so called hard and soft errors. If a hard error exists the output ERR is set to 1. No updating of the outputs is then performed, excepting possible default values. With soft errors, the element is executed normally. For both types of errors the output ERRTYPE is set to a value according to the table below. If several errors exists simultaneously, the error code that occurs first in the table is shown and the next remaining error is shown when the first one goes away.

Error Codes

Error	ERR=1	Error description
100	X	Internal data base error.
1	X	The data base element given by the parameter UNITNO on the PC element is missing.
2	X	The data base element given by the parameter UNITNO on the PC element is not of the type PX.
3	X	The data base element is not accepted. This may be caused by the parameter IMPL being 0 or that the bus number is erroneous.
4	X	The bus controller module is missing or faulty.
5	X	MasterPiece 51 is missing on the bus (communication error).
6	X	Initialization in progress. If the indication persists it indicates a system malfunction.

Error Codes

Error	ERR=1	Error description
7	X	The basic unit connected is not a MasterPiece 51.
8		The wrong type of expansion unit is connected.
9		Error in the transducer supply of the basic unit.
10		Error in the transducer supply of the expansion unit.
11	X	No local application program is running in the MasterPiece 51.
12		Execution in MasterPiece 51 has been stopped at a breakpoint.
13		The MasterPiece 51 is in local mode and receives no information from the controller.
14		One or more of MasterPiece 51's I or Q channels are force set.
15		One or more of MasterPiece 51's IR or QR channels are force set.
16		Low battery voltage in MasterPiece 51 or battery is missing.
17	X	Reset-execution. The QR outputs are frozen. The element is not executed.
18		The QR outputs of the PC element have DEFAULT values.
0	--	OK



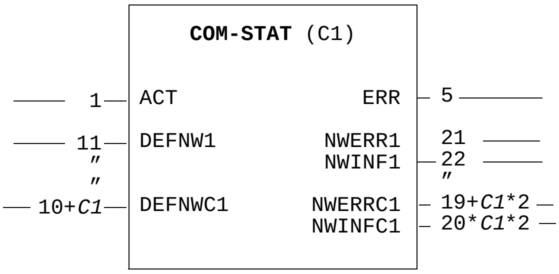
Function diagram

Communication Supervision COM-STAT

Summary

Supervision of networks with belonging nodes which are directly connected to the controller. The following links are supervised:

- MasterBus 300/300E (MB300)
- GCOM
- RCOM

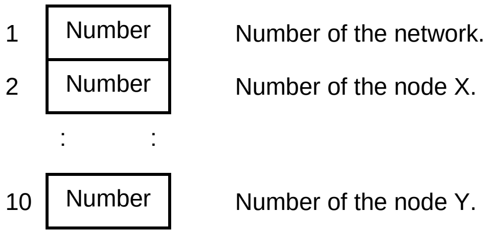


PC element COM-STAT

Call COM-STAT (C1)

Parameter	Significance	Permissiblevalues
C1	C1 = 1 - 9. Status of C1 number of networks with belonging nodes, which the controller node is directly connected to.	1 - 9, default 1

Terminal Description

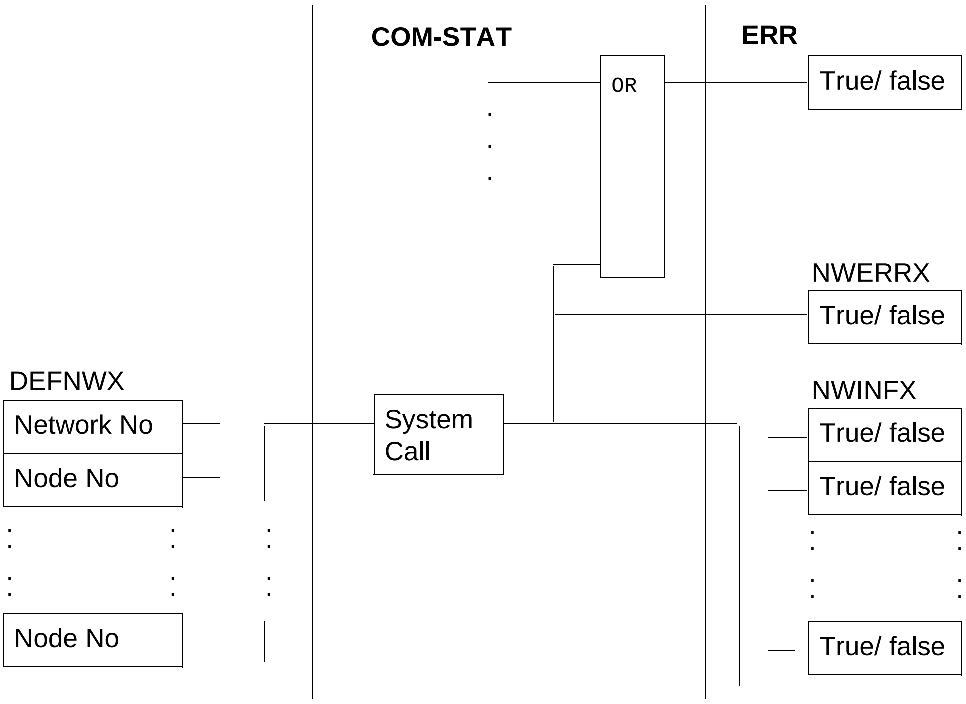
No	Name	Type	Description
1	ACT	IB	**ACT**ivate, input for activation of the function.
5	ERR	OB	**ERR**or, general signal to indicate error on the network.
11	DEFNW1	IG10I	**DEF**inition of **N**et**W**ork, definition of network and belonging nodes whose status are supervised. The status is given in NWINF1. In not used positions of DEFNW1 is (0) zero written.  1 Number Number of the network. 2 Number Number of the node X. : : 10 Number Number of the node Y.

Terminal Description

No	Name	Type	Description
10+*C2*	DEFNW*C2*	IG10I	**DEF**inition of **N**et**W**ork and belonging nodes which status are supervised. The status is given in NWINF*C2*. Zero (0) is written in not used positions of DEFNW*C2*. 1 Number Number of the network. 2 Number Number of the node Z. : : 10 Number Number of the node W.
21	NWERR1	OB	**N**et**W**ork **ERR**or, error in the network or/and nodes.
22	NWINF1	OG10B	**N**et**W**ork **INF**ormation, information about network and nodes defined in DEFNW1. 1 True/ false Error in network. 2 True/ false Error in node X. : : 10 True/ false Error in node Y.
: 19+2*C2	NWERR*C2*	OB	**N**et**W**ork **ERR**or, error in the network or/and nodes defined by DEFNW*C2*.
20+2*C2	NWINF*C2*	OG10B	**N**et**W**ork **INF**ormation, information about network and nodes defined in DEFNW*C2*. 1 True/ false Error in network. 2 True/ false Error in node X. : : 10 True/ false Error in node Y.

Function

DEFNWX (X=1 - 9) is a table with ten positions where the first one is the number of the supervised network and in following positions belonging nodes which are supervised. Positions which are not used are set to zero. In NWINFX, a corresponding table to DEFNWX, the malfunction is indicated. NWERRX (X=1 - 9) is a summary signal for NWINFX.



Links of communication

Restrictions

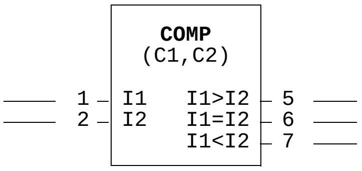
The PC element should be executed with a cycle time of at least 1 second, to make sure that the execution of COM-STAT gives low processor load.

Comparator COMP

Summary

COMP (**COMP**arator) is used to compare two values.

The data type may be integers, real numbers, Boolean values, time or character string.



PC element COMP

Call COMP (C1, C2)

Parameter	Significance	Permissible value
C1	Data type	I, IL, R, B, T, TR, A
C2	Number of characters for data type A	0 - 80, default 0

Terminal Description

No	Name	Type	Description
1	I1	I*C1*	**I**nput **1**. Input for value 1.
2	I2	I*C1*	**I**nput **2**. Input for value 2.
5	I1>I2	OB	**I**nput **1>I**nput **2**. Output which is set to 1 if the value at I1 is greater than the value at I2.
6	I1=I2	OB	**I**nput **1=I**nput **2**. Output which is set 1 if the value at I1 is equal to the value at I2.
7	I1<I2	OB	**I**nput **1<I**nput **2**. Output which is set to 1 if the value at I1 is less than the value at I2.

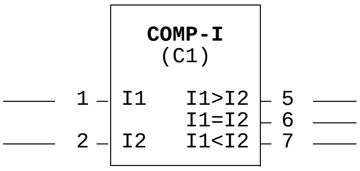
Function

The values at the two inputs are compared and the result can be read at the outputs I1>I2, I1=I2 and I1<I2. If the data type is A (character string), the two strings are compared, character by character, until they differ in a position. The result is then determined by the comparison between these two characters. If no difference is found the output I1=I2 is set to 1. At startup the output I1=I2 is set before the first execution, independent of the values of inputs I1 and I2.

Comparator for Integers COMP-I

Summary

COMP-I (**COMP**arator **- I**nteger) is used to compare two integers.



PC element COMP-I

Call COMP-I (C1)

Parameter	Significance	Permissible values
C1	Data type	I, IL

Terminal Description

No	Name	Type	Description
1	I1	I*C1*	**I**nput **1**. Input, the value of which is to be compared with input I2.
2	I2	I*C1*	**I**nput **2**. Input, the value of which is to be compared with input I1.
5	I1>I2	OB	**I**nput **1**>**I**nput **2**. Output which is set if the value at the input I1 is greater than the value at input I2.
6	I1=I2	OB	**I**nput **1**=**I**nput **2**. Output which is set if the value at the input I1 is equal to the value at input I2.
7	I1<I2	OB	**I**nput **1**<**I**nput **2**. Output which is set if the value at the input I1 is less than the value at input I2.

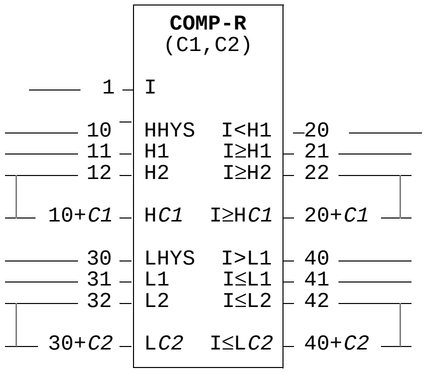
Function

The values at the two inputs are compared and the result of the comparison can be read at the outputs I1<I2, I1=I2 and I1>I2. At startup, the output I1=I2 is set before the first execution independent of the values of inputs I1 and I2.

Comparator for Real Numbers COMP-R

Summary

COMP-R (**COMP**arator **- R**eal) is used for limit value monitoring of real numbers with relation to several limits.



PC element COMP-R

Call COMP-R (C1, C2)

Parameter	Significance	Permissible values
C1	Number of limit values for high level	1 - 9
C2	Number of limit values for low level	1 - 9

Terminal Description

No	Name	Type	Description
1	I	IR	**I**nput. Input, the value of which is to be compared with limit values H1 - H*C1* and L1 - L*C2*.
10	HHYS	IR	**H**igh **HYS**teresis. Hysteresis common for all H values. The hysteresis is specified as the difference between the I-value at which the H outputs are set and the value at which they are reset.
11	H1	IR	**H**igh **1**. Upper limit value 1.
12	H2	IR	**H**igh **2**. Upper limit value 2.
: 10+*C1*	H*C1*	IR	**H**igh ***C1***. Upper limit value *C1*.
20	I<H1	OB	**I**nput<**H1**. Output which is set if I is less than H1.
21	I≥H1	OB	**I**nput≥**H1**. Output which is set if I is greater than or equal to H1.
22	I≥H2	OB	**I**nput≥**H2**. Output which is set if I is greater than or equal to H2.
: 20+*C1*	I≥H*C1*	OB	**I**nput≥**H***C1*. Output which is set if I is greater than or equal to H*C1*.
30	LHYS	IR	**L**ow **HYS**teresis. Hysteresis common for all L values. The hysteresis is given as the difference between the I- value at which the L outputs are reset and the value at which they are set.
31	L1	IR	**L**ow **1**. Lower limit value 1.

Terminal Description

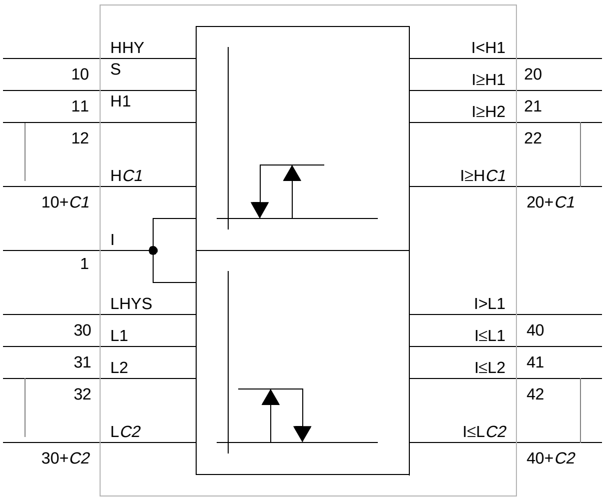
No	Name	Type	Description
32	L2	IR	**L**ow **2**. Lower limit value 2.
: 30+*C2*	L*C2*	IR	**L**ow ***C2***. Lower limit value *C2.*
40	I>L1	OB	**I**nput>**L1**. Output which is set if I is greater than L1.
41	I≤L1	OB	**I**nput≤**L1**. Output which is set if I is less than or equal to L1.
42	I≤L2	OB	**I**nput≤**L2**. Output which is set if I is less than or equal to L2.
: 10+*C2*	I≤L*C2*	OB	Input≤L*C2*. Output which is set if I is less than or equal to L*C2.*

Function

Input signal I is compared with the limit value specified at the inputs H1 - H*C1* and L1 - L*C2*. For upper limits, the output for the different limits will be set when the input I becomes equal to, or greater than, the limit value. For lower limits, the output is set when I becomes less than or equal to the limit value. Limits can be set optionally within the interval $-9.2x10^{18}$ .. $9.2x10^{18}$. The lowest representable positive and negative values are $5.0x10^{-20}$ and $–5.0x10^{-20}$. The element put outputs 20 (I<H1) and 40 (I>L1) to 1 before the first execution independent of input I.

Hysteresis

The input HHYS gives the hysteresis for all limits at high level and LHYS for all limits at low level. The hysteresis is given as the difference between the I values for which the different outputs are set and the values for which they are reset. At high level the outputs will be reset when I becomes lower than the limit minus the hysteresis HHYS. At low level the outputs will be reset when I becomes greater than the limit value plus the hysteresis LHYS.

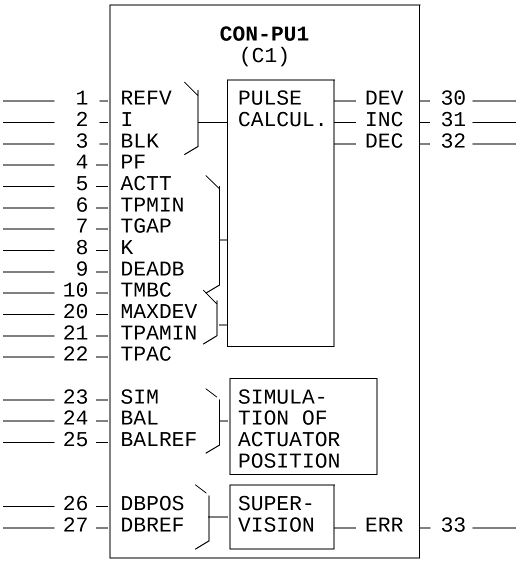


Function diagram

Three State Controller CON-PU1

Summary

CON-PU1 (**CON**troller **- PU**lsed **1**) is a three state controller intended to control, by means of pulses (increase/decrease), a process via an integrating actuator such as a shifting motor or a solenoid valve. CON-PU1 can be combined with, for example, a PI regulator for process control. If only pulse generation is desired, a simplified function can be chosen with a call parameter.



PC element CON-PU1

Call CON-PU1 (C1)

Parameter	Significance	Permissible values
C1	Selection of extended or ordinary function. The value 0 gives ordinary function.	0 - 1

Terminal Description

No	Name	Type	Description
1	REFV	IR	**REF**erence **V**alue. Input for setpoint.
2	I	IR	**I**nput value. Input for measured value.
3	BLK	IB	**BL**oc**K** pulse calculation.
4	PF	II	**P**ulse **F**requency Factor. Minimum pulse cycle time=PF x TS.
5	ACTT	ITR	**ACT**uator **T**ime. The delay time interval of the actuator.
6	TPMIN	ITR	**T**ime **P**ulse **MIN**imum. Min pulse length for excitation of the motor.
7	TGAP	ITR	**T**ime **GAP**. Compensation for play.
8	K	IR	Pulse length factor. Pulse=(KxDEVxACTT) /100+TGAP+TPMIN.
9	DEADB	IR	**DEADB**and. $\lvert DEV \rvert \geq DEADB$ if a pulse is to be given.
10	TMBC	ITR	**T**ime **M**inimum **B**etween **C**hange. Minimum pause time after DEV having changed sign.
20(1)	MAXDEV	IR	**MAX**imum **DEV**iation for pulse length calculation. This value does not affect output DEV.
21(1)	TPAMIN	ITR	**T**ime **PA**use **MIN**imum. Minimum pause time.
22(1)	TPAC	IR	**T**ime **PA**use **C**onstant. Pause Time factor. Pulse time=TPAMIN+TPAC x Pulse duration.
23(1)	SIM	IB	**SIM**ulate actuator position. Input for activation of simulation of actuator position.
24(1)	BAL	IB	**BAL**ance. Input for activation of tracking.
25(1)	BALREF	IR	**BAL**ance **REF**erence. Input for reference value when tracking.

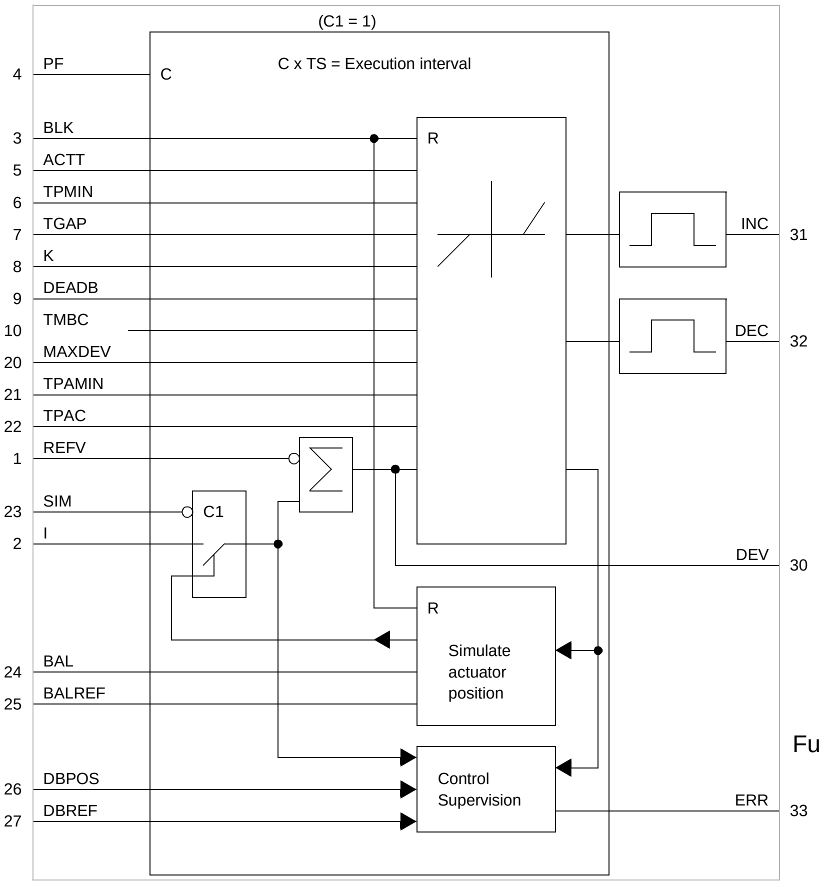
Terminal Description

No	Name	Type	Description
26(1)	DBPOS	IR	**D**ead**B**and for actuator **POS**ition change (in %).
27(1)	DBREF	IR	**D**ead**B**and **REF**erence for position change alarm. When actuator position is calculated to have changed > DBPOS, a check is made to determine if | I(t=n) - I(t=0) | < DBREF, if so ERR is set, otherwise ERR is cleared. I(t=0) is the Input value when the calculation started. I(t=n) is the Input value when calculated actuator position has changed > DBPOS.
30	DEV	OR	Position **DEV**iation. DEV=I - REFV.
31	INC	OB	**INC**rease. Output for increase of actuator position.
32	DEC	OB	**DEC**rease. Output for decrease of actuator position.
33(1)	ERR	OB	**ERR**or. Actuator position has not followed the change commanded.

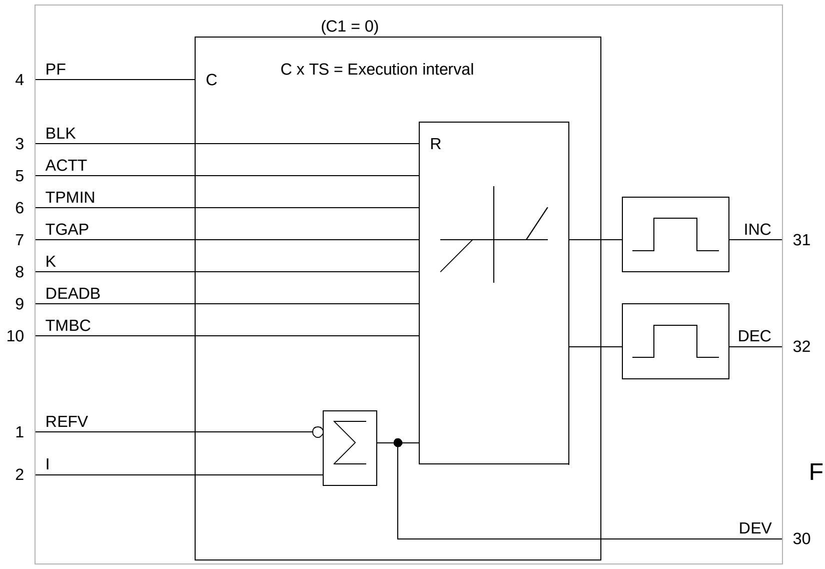
(1) Only used with extended function, (*C1*=1)

Function

The three state controller CON-PU1 is intended to control, by means of pulses at the outputs INC and DEC (increase/decrease respectively), a process via, for example, a shifting motor or a solenoid valve. With the call parameter *C1* can be selected ordinary (*C1*=0) or extended (*C1*=1) function. The following inputs should be in unit % (0-100): REFV, I, DEADB, MAXDEV, BALREF, DBPOS, DBREF.



Function diagram for extended function (C1=1)



Function diagram for ordinary function (C1=0)

Extended Function (C1=1)

The feedback can be obtained from the actuator or the element can simulate the actuator position internally. The effective pulse duration is proportional to the control error and is calculated as follows:

$PL_{eff}=| DEVxKxACTT | / 100$, where:

DEV=Control deviation=I–REFV (%)

K=Pulse length factor

ACTT=Actuator delay time interval in seconds

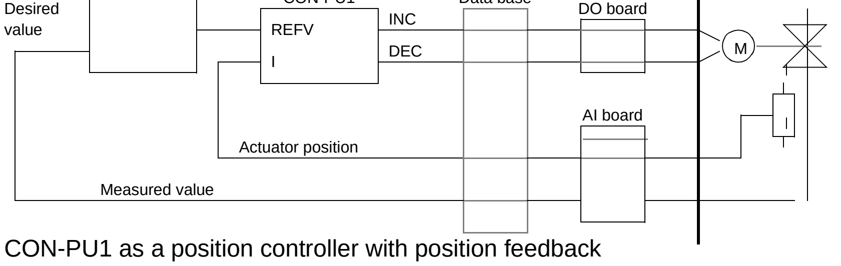
To permit gradual correction of a large control error, |DEV|, is limited to MAXDEV. The actual time pulse duration is calculated as:

PL=(| DEVxKxACTT |)/100+TGAP+TPMIN where:

TPMIN=Min pulse duration for excitation of the shifting motor or solenoid valve. This is the shortest pulse length which affects the actuator.

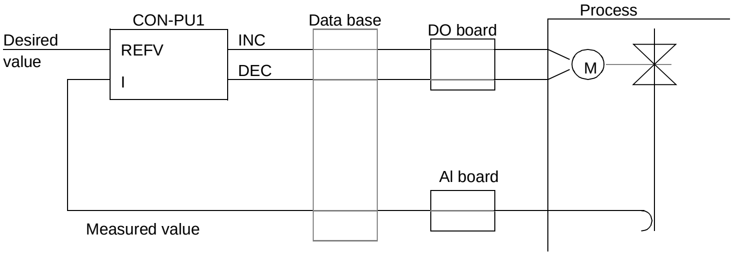
TGAP=Play compensation. Added to the first pulse after a polarity change.

For control errors less than a deadband (DEADB, %) the correction is considered to be complete and no further pulse is calculated. TMBC is a delay time that gives the shortest pause time after a change of the pulse outputs INC/DEC. Calculation of a new pulse duration is activated at the interval PF x TS where TS is the sample time of the function, that is, the cycle time of the element.

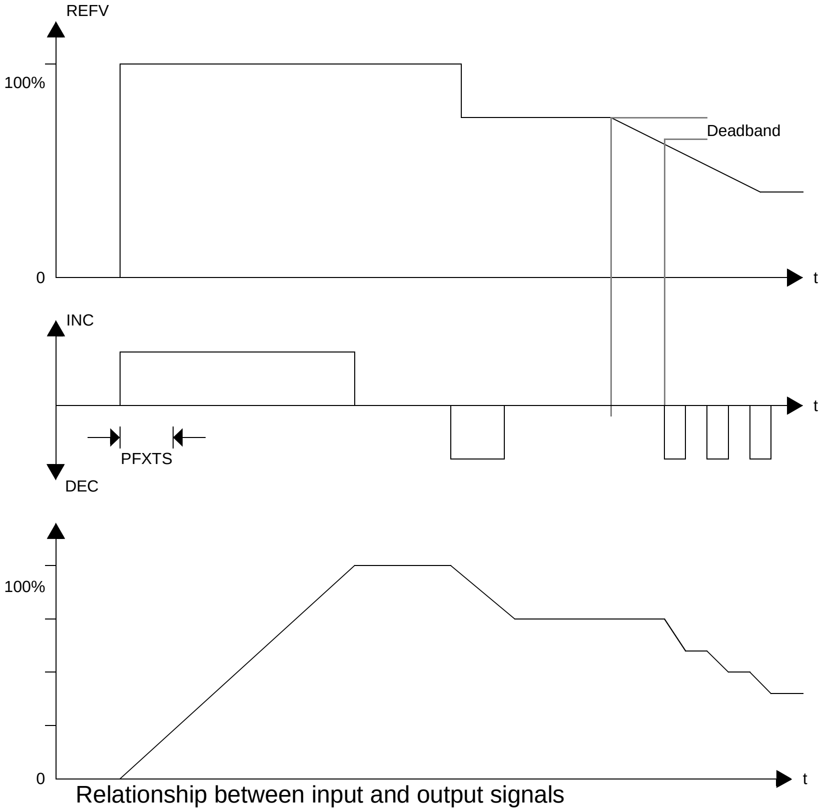


CON-PU1 as a position controller with position feedback

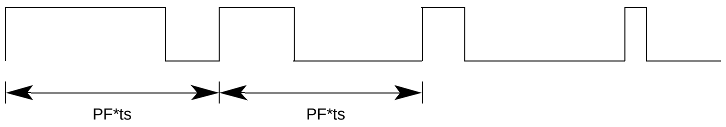
The least permitted pause time between two pulses is determined however by the parameters TPAMIN and TPAC so that the Pause time=TPAMIN+TPAC x PL where PL is the pulse duration calculated most recently. These possibilities of setting the pause time are normally only used when CON-PU1 is used as a stand alone regulator and provide great flexibility in trimming the regulator. By setting PF x TS relatively large and TPAMIN=TPAC=0, a constant pulse frequency is obtained. To obtain a constant pause time between the pulses, TPAC is set to 0 and TPAMIN to the required pause time. This is necessary when controlling processes with long time constants.



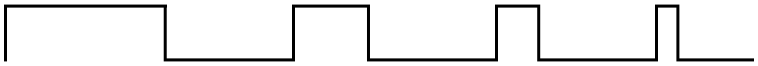
CON-PU1 as a stand alone controller



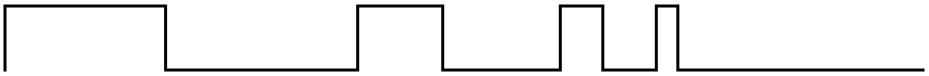
Relationship between input and output signals



Pause time = TPAMIN + TPAC x PL, TPAMN =TPAC = 0



Pause time = TPAMIN + TPAC x PL, TPAC = 0



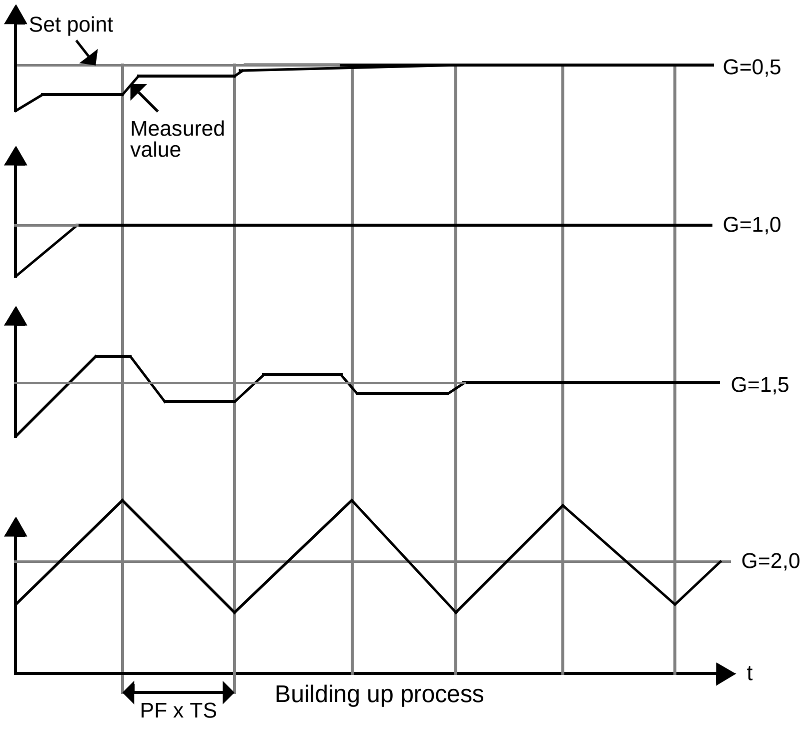
Pause time=TPAMIN + TPAC x PL. TPAMIN and TPAC > > 0

In processes with long delay times, pause times ≥ the delay time are selected.

The pause time can be made to be dependent on the duration of the latest pulse calculated with the parameter TPAC.
This function is used when a rapid control of small errors is required but when limits on the speed of change in the process response require long pause times with large changes.

When CON-PU1 is used as a position regulator for a motor actuator, different building up processes are obtained depending on a parameter G, this being in principle the relationship between the real data of the actuator and the parameters set in CON-PU1.

G=(kxACTT)/ACTT$_v$+100x(TGAPzTGAP$_v$+TPMINzTPMIN$_v$)/ (ACTT$_v$x│ DEV │); TGAP$_v$ is the actuator play, TPMIN$_v$ the least pulse length to which the actuator reacts, and ACTT$_v$ the time delay time interval of the actuator. The figure below shows the building up process for different G, where only (k x ACTT)/ACTT$_v$ is included. The system is unstable when G≥2



Simulation of Actuator Position

If it is impossible to feed back the actuator position to the element, it can be simulated internally in the element by integrating DEV x K x ACTT. The simulated value is limited to 0-100%. When BAL is set, BALREF is used as an integrator value, for example, with a limit switch on the actuator to set the integrator to a known value or forced control when controlling the process manually. With simulation, the preceding regulator should have an output signal >0–100% (–100–200%).

Supervision

When actuator position is calculated to have changed > DBPOS, a check is made to determine if | I(t=n) - I(t=0) | < DBREF, if so, ERR is set, otherwise ERR is cleared.

I(t=0) is the input value when the calculation started and I(t=n) is the input value at the time when calculated actuator position has changed > DBPOS. DBPOS is given in % of the span the actuator is to have moved in ACTT seconds. The calculation starts all over after exceeding DBPOS, when DEV changes sign or when DBPOS is not exceeded after 9 pulses.
ERR is reset when DEV changes sign, or if DBPOS is not exceeded after 9 pulses.
Theoretically you can set DBREF=DBPOS. This means that the actuator follows the controller without delay. In practice, you will have to set DBREF<DBPOS.

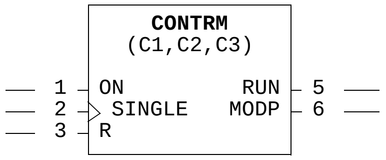
Ordinary Function (C1=0)

With ordinary function only pulse generation according to PL= (│DEV│ xKIxACTT)/100+TPMIN+TGAP is performed.
If │DEV│ <DEADB no pulse is obtained.

Control Module Header CONTRM

Summary

CONTRM (**CONTR**ol **M**odule Header) is used for structuring and execution control of control modules. Function modules, blocks and elements can be included in a control module.



PC element CONTRM

Call CONTRM (C1, C2, C3)

Parameter	Significance	Permissible values
C1	Cyclicity in ms	Permissible values are determined by the possible cycle times of the relevant PC system. These times are specified in the manual for the system concerned.
C2	Place in the cycle time table	Permissible values are determined by the relation to other execution units with the same cycle time. The max value is determined by the number of possible places with the same cycle time. The system selects a free place with the lowest number if no value is specified at entry.
C3	Currently not used	

Terminal Description

No	Name	Type	Description
1	ON	IB	Control input which is to be set (to 1) with normal execution.
2	SINGLE	IB	If input ON and R are 0 the elements in the control module can be executed once by setting **SINGLE**.
3	R	IB	**R**eset. Input for clearing of the elements in the control module.
5	RUN	OB	**RUN** indicates that the elements in the control module are executed.
6	MODP	OB	**MOD**ify **P**ermission. Output that indicates if the module may be modified from service aids.

Function

The control module header CONTRM is a superior execution controlling element for a control module. The call parameter *C1* is used to specify how often the control module is to be executed and call parameter *C2* is used to specify the place in the cycle time table on which the execution sequence of execution units with the same cycle time is based. When and how the control module is to be executed is determined with the control inputs ON, SINGLE and R.

Normal Execution

The input ON is to be set with normal execution and the control module is then executed in accordance with the call parameters *C1* and *C2*. If the ON input is reset after execution of the control module, the calculated data remains until the next time the control module is executed. The ON input overrides the SINGLE input, that is, if ON is set SINGLE has no effect.
The execution is performed in three steps:

1. Entry of variables from the data base and from modules with higher priority.
2. Element execution.
3. Reading of variables to the data base and to modules with higher priority.

SINGLE Execution

If the SINGLE input is set when the ON input is reset, the control module is executed once in accordance with normal execution.

Clearing

If input R is set, the control module is executed in the reset mode. This means that all outputs of the elements within the control module are given original values which in most cases are the 0-value of the data. Reading to the data base is performed with data value=0 for all data types. Input R overrides the inputs ON and SINGLE.

RUN

Output RUN is set only if normal execution is in progress, that is, if ON is set, or if the control module is being SINGLE executed. With SINGLE execution, the RUN output is set during one cycle only.

MODP

The control module header contains a built in check of the right to change, MODP. If the changing right is deactivated (MODP=0), the possibility of introducing changes in the control module function from service aids is blocked. However, blocking can still be entered and operational parameters can be changed. The changing right is activated and deactivated from service aids with special commands.

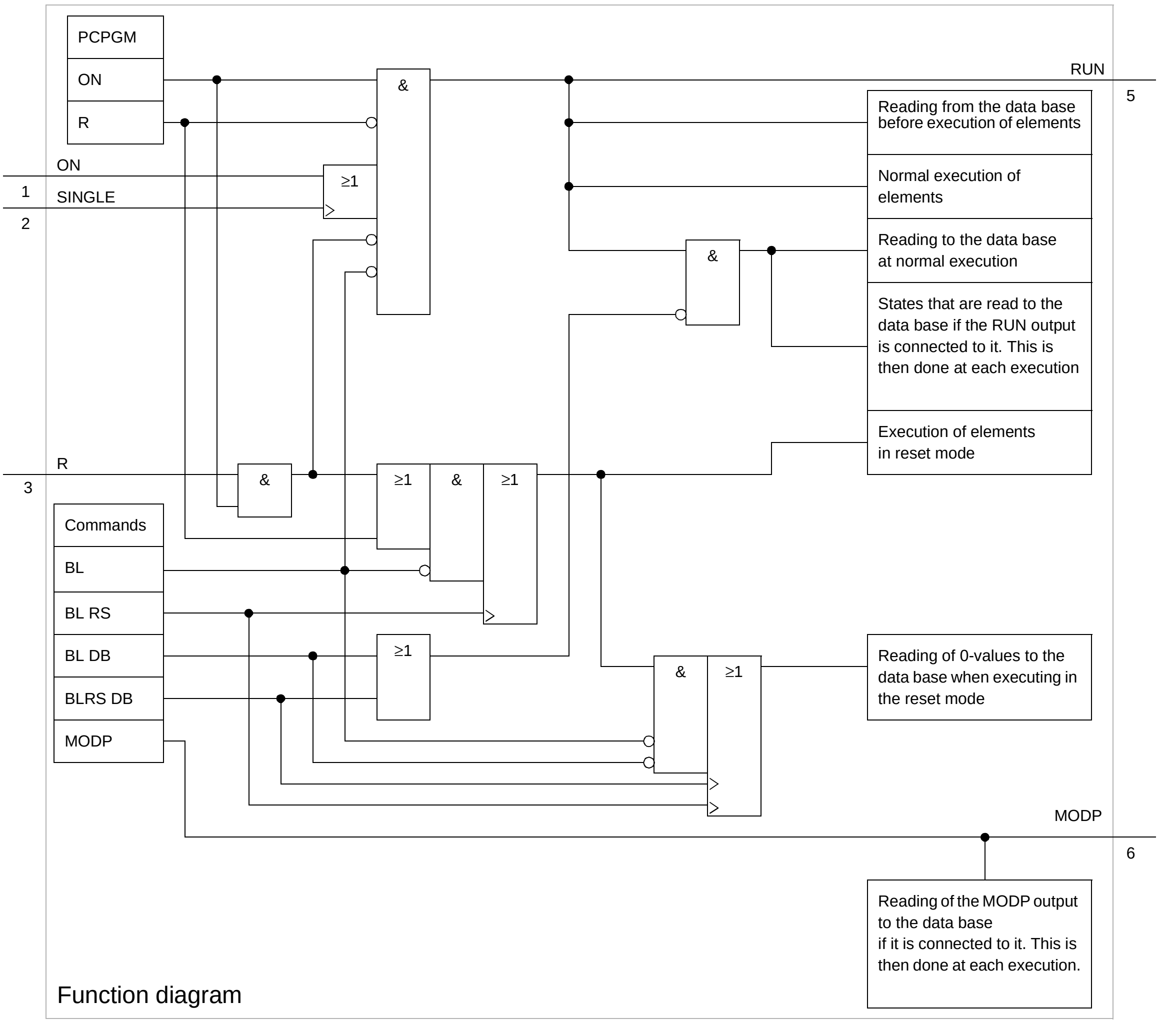
Blocking

The execution of a control module can be blocked in different ways from a service aid.

- All execution of the control module including data exchange with the data base is blocked with the command BL<module identity>(**BL**ock).
- Reading to the data base of the data base variables included in the control module is blocked with the command BL DB,<module identity>(**BL**ock **D**ata**B**ase).
- The control module and data to the data base are first cleared (Reset execution) and execution is then blocked with the command BLRS<module identity>(**BL**ock and **R**e**S**et).
- Output data to the data base for the control module is first cleared after which further reading is blocked by means of the command BLRS DB,<module identity>(**BL**ock and **R**e**S**et **D**ata**B**ase).
- Blocking can be removed with the command DBL<module identity>(**D**e**BL**ock).

Effects from PCPGM

ON and the R inputs and the blocking functions on the program header override the inputs on the control module header. The input ON on PCPGM must be set to permit normal execution or reset execution of the elements in the control module.

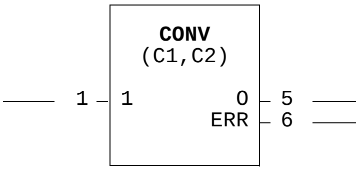

PCPGM
ON
R
RUN
5
&
Reading from the data base before execution of elements
ON
1
SINGLE
2
≥1
Normal execution of elements
Reading to the data base at normal execution
&
States that are read to the data base if the RUN output is connected to it. This is then done at each execution
Execution of elements in reset mode
R
3
&
≥1
&
≥1
Commands
BL
BL RS
BL DB
BLRS DB
MODP
≥1
&
≥1
Reading of 0-values to the data base when executing in the reset mode
MODP
6
Reading of the MODP output to the data base if it is connected to it. This is then done at each execution.

Function diagram

Code Converter CONV

Summary

CONV (**CONV**erter) converts data from and to integers, real numbers and time.



PC element CONV

Call CONV (C1, C2)

Parameter	Significance	Permissible values
C1	Data type at input	I, IL, R, T, TR
C2	Data type at output	I, IL, R, T, TR

Terminal Description

No	Name	Type	Description
1	I	I*C1*	**I**nput. Input for data to be converted.
5	O	O*C2*	**O**utput. Output for converted data.
6	ERR	OB	**ERR**or. Output which is set (to 1) if a limit is reached at output O.

Function

Data at input I with data type in accordance with call parameter *C1* is converted to data with data according to call parameter *C2* at the output O. When conversion is to the data types I and IL (integer) and T (time), overflow can occur at the max or min limit values. The error signal output ERR is then set. Conversion from T to I or IL gives number of 1/10 ms. Conversion from T to R gives number of S.

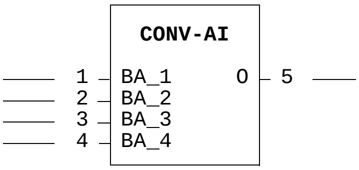
Rounding off

When converting from real numbers to integers, the decimal part of the number to be converted is rounded off downwards in the interval 0 .. 0.499 ... and upwards in the interval 0.5 .. 1. When converting from T to R, I or IL the value given is the number of **entire** seconds from midnight. When converting from R, I or IL to T the input value is treated as the number of **entire** seconds from midnight.

Code Converter, Array to Integer CONV-AI

Summary

CONV-AI (**CONV**erter **- A**rray to **I**nteger) is used to convert four character strings, each of length 1, to a long integer (IL). The element is used to send characters to MasterView 320 via the data base.



PC element CONV-AI

Call CONV-AI

Terminal Description

No	Name	Type	Description
1	BA_1	IA1	**B**yte **A**rray **1**. Input for character corresponding to the lowest byte of O.
2	BA_2	IA1	**B**yte **A**rray **2**. Input for character corresponding to the second byte of O.
3	BA_3	IA1	**B**yte **A**rray **3**. Input for character corresponding to the third byte of O.
4	BA_4	IA1	**B**yte **A**rray **4**. Input for character corresponding to the highest byte of O.
5	O	OIL	**O**utput. Output for long integer constructed from BA1 .. BA4.

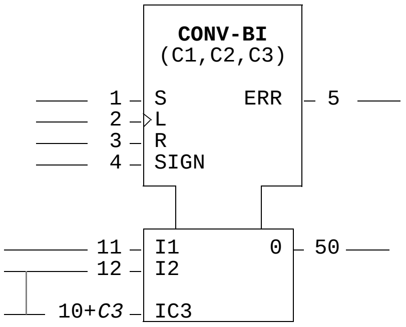
Function

CONV-AI forms a long integer from four bytes. If a byte corresponds to a non printable character, 0 (NUL) is used as a substitute for the character.

Code Converter, Boolean to Integer CONV-BI

Summary

CONV-BI (**CONV**erter **- B**oolean to **I**nteger) converts data from BC, BCD, 1-of-N, Gray code or 16/32 Bit Packing, formed from Boolean variables into integers. The element has a memory function for storage of converted data.



PC element CONV-BI

Call CONV-BI (C1, C2, C3)

Parameter	Significance	Permissible values
C1	Data type of output	I, IL
C2	Type of conversion 1 for BC 2 for BCD 3 for 1-of-N code 4 for Gray code 5 for Bit Packing(1)	1 - 5
C3	Number of inputs for data to be converted. 1 .. 31 If C2=5 then C3 must be 15 or 31.	1 - 31

(1) When C2=5 the output O cannot be interpreted as I/IL by the aid.

Number of inputs required with

Code	Number of inputs required with	
	I	**IL**
BC	15	31
BCD	18	31
1-of-N	31	31
Gray	15	31
Bit packing	15+ SIGN	31+SIGN

Terminal Description

No	Name	Type	Description
1	S	IB	**S**et. Input for storage of a new value each time the element is executed. When this input is reset (to 0) the latest calculated value will remain at the outputs.
2	L	IB	**L**oad. Dynamic input for loading of data.
3	R	IB	**R**eset. Input for clearing of the output. This input overrides the S and L inputs.

Terminal Description

No	Name	Type	Description
4	SIGN	IB	Input set (to 1) with negative data values. If C2=5 SIGN is used as the most significant input to get a 16/32 bit conversion.
5	ERR	OB	**ERR**or. Output which is set when more than one input is set with conversion of 1-of-N code or if the integer value to be stored at the output cannot be represented by the data type.
11	I1	IB(1)	**I**nput **1**.
12	I2	IB(1)	**I**nput **2**.
: 10+*C3*	I*C3*	IB(1)	**I**nput ***C3***.
50	O	O*C1*	**O**utput. Output for converted data.

(1) The data at each of these inputs is of Boolean type but together they form a value of the type specified by parameter *C1*.

Function

Converts binary code (BC), binary coded decimal code (BCD), 1-of-N code, Gray code, or uncoded Booleans (Bit Packing) to integers of type I (16 bit) or IL (32 bit). SIGN input can be used as a sign bit (C2=1, 2 or 3) or as most significant bit (C2=5).

Example of different decodings

Input terminal	Output Value O			
	C2=1, BC	C2=2, BCD	C2=3, 1-of-N(1)	C2=5, Packing
I1	1	1	1	1
I2	2	2	2	2
I3	4	4	3	4
I4	8	8	4	8
I5	16	10	5	16
I6	32	20	6	32
I7	64	40	7	64
I8	128	80	8	128
: I31	: 2^{30}	: 4×10^7	: 31	: 2^{30}
SIGN	* -1	* -1	* -1	2^{31}

(1) Only one of the inputs may be set with 1-of-N code.

Example of Graycode decoding (C2=4)

Inputs				Output Value O
I4	I3	I2	I1	
0	0	0	0	0
0	0	0	1	1
0	0	1	1	2
0	0	1	0	3
0	1	1	0	4

Example of Graycode decoding (C2=4)

Inputs				Output Value O
I4	**I3**	**I2**	**I1**	
0	1	1	1	5
0	1	0	1	6
0	1	0	0	7
1	1	0	0	8
1	1	0	1	9
1	1	1	1	10
1	1	1	0	11
1	0	1	0	12
1	0	1	1	13
1	0	0	1	14
1	0	0	0	15

Storage of Data

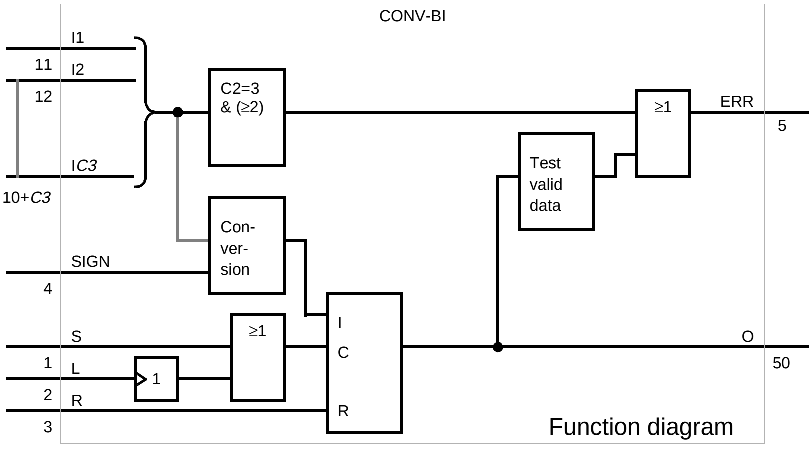
The code at the inputs I1 .. I*C3* is converted and stored immediately the input L is set. If input S is set, the input code is converted and the integer is stored each time the element is executed. When S is reset (to 0) after having been set, the data stored most recently remains. The input S overrides the input L, that, is when S is set, L has no effect.

Clearing

The input R clears the output and prevents all further storage of data while R is set.

Supervision

When converting 1-of-N code the setting of only one of the inputs 1 .. I*C3* is supervised. If two or more inputs are set, the value of the input signal with the lowest number is stored. The error signal output ERR is also set. If the integer value to be stored at the output exceeds the value which can be represented, the error signal output is set. In addition, the output is limited to the upper or lower limit value.

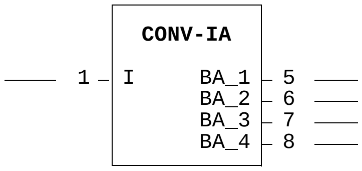


Function diagram

Code Converter, Integer to Array CONV-IA

Summary

CONV-IA (**CONV**erter **- I**nteger to **A**rray) is used to convert a long integer (IL) to four character strings each of length 1. The element is used to read characters to MasterView 320 via the data base.



PC element CONV-IA

Call CONV-IA

Terminal Description

No	Name	Type	Description
1	I	IIL	**I**nput. Input for long integer to be split up.
5	BA_1	OA1	**B**yte **A**rray **1**. Output for character corresponding to the lowest byte of I.
6	BA_2	OA1	**B**yte **A**rray **2**. Output for character corresponding to the second byte of I.
7	BA_3	OA1	**B**yte **A**rray **3**. Output for character corresponding to the third byte of I.
8	BA_4	OA1	**B**yte **A**rray **4**. Output for character corresponding to the highest byte of I.

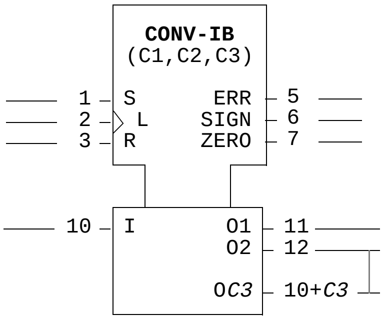
Function

CONV-IA splits a long integer into four bytes. If a byte corresponds to a non printable character, ”space” (H'20) is substituted for the character.

Code Converter, Integer to Boolean CONV-IB

Summary

CONV-IB (**CONV**erter **- I**nteger to **B**oolean) converts data from integer to Boolean variables using BC, BCD, 1-of-N, conversion or uncoded unpacking. The element has a memory function for storage of converted data.



PC element CONV-IB

Call CONV-IB (C1, C2, C3)

Parameter	Significance	Permissible values
C1	Data type	I, IL
C2	Type of conversion 1 for BC 2 for BCD 3 for 1-of-N code 4 for bit unpacking	1 - 4
C3	Number of converted data.	1 - 32

Number of Outputs Required with Different Codes

Code	Number of outputs required with	
	I	IL
BC	15	31
BCD	18	32
1-of-N	31	32
Bit packing	15+ SIGN	31+SIGN

Terminal Description

No	Name	Type	Description
1	S	IB	**S**et. Input for storage of a new value each time the element is executed. When this input is reset (to 0) the latest calculated value will remain at the outputs.
2	L	IB	**L**oad. Dynamic input for loading of data.
3	R	IB	**R**eset. Input for clearing of the outputs. This input overrides the S and L inputs.
5	ERR	OB	**ERR**or. Output which is set when the integer value to be converted exceeds the value which can be represented at the outputs.
6	SIGN	OB	Output set to 1 with negative data values. If C2=4 SIGN is interpreted as the most significant bit.
7	ZERO	OB	Output set to 1 when the value at the input I is zero, or when input R is set to 1.
10	I	IC1	**I**nput. Input for the integer value to be converted.
11	O1	OB(1)	**O**utput **1**.
12	O2	OB(1)	**O**utput **2**.
: 10+*C3*	O*C3*	OB(1)	**O**utput ***C3***.

(1) The data at each of these outputs is of Boolean type but together they form a value of the type specified by parameter C2.

Function

CONV-BI converts integer values of data types I or IL into Boolean variables, using binary code (BC), binary coded decimal code (BCD), 1-of-N code, or uncoded unpacking. The code conversion to be performed is specified by the call parameters *C1* and C2. With positive data values, the output SIGN is reset (to 0). With negative values, it is set (to 1). For bit unpacking, SIGN is used as the most significant bit.

Examples of Output Values with Different Decodings

Value of Input I				Affected Output
C2= 1 BC	C2= 2 BCD	C2= 3 1-of-N	C2 =4 Unpack	
1	1	1	1	O1
2	2	2	2	O2
4	4	3	4	O3
8	8	4	8	O4
16	10	5	16	O5
32	20	6	32	O6
64	40	7	64	O7
128	80	8	128	O8
256	100	9	256	O9

Examples of Output Values with Different Decodings

Value of Input I				Affected Output
C2= 1 BC	C2= 2 BCD	C2= 3 1-of-N	C2 =4 Unpack	
512	200	10	512	O10
: 2^{30}	: 4×10^{7}	: 31	: 2^{30}	: O31
-	4×10^{7}	32	-	O32
(1)	(1)	(1)	2^{31}	SIGN

(1) SIGN is used as a sign for BC, BCD and 1-of-N.
Only one of the outputs is set with 1-of-N code.

Storage of Data

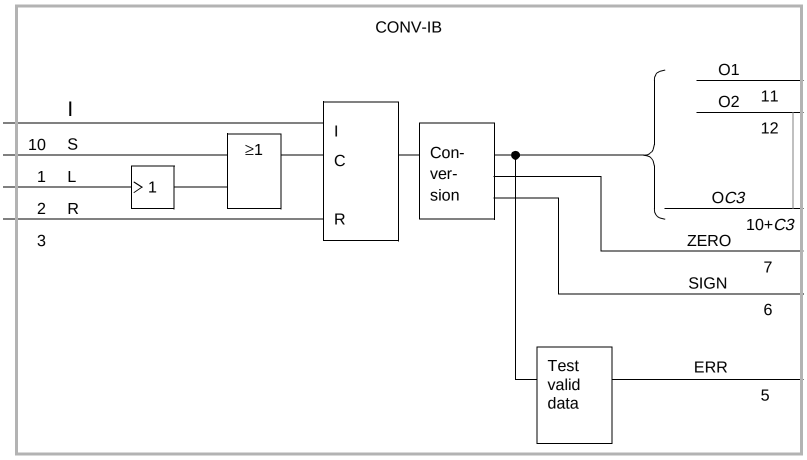
The integer at input I is converted immediately when input L is set. If input S is set, the input code is stored, and the result is converted each time the element is executed. When S is reset (to 0), after having been set, the data stored most recently remains until the element once more is executed with one of the inputs S, L or R set. The input S overrides the input L, that is, when S is set, L has no effect.

Reset

The input R clears the data outputs O1 to O*C3*, and prevents all further storage of data while R is set.

Supervision

When converting data from integers to 1-of-N code, the numbers are restricted to –32 to +32. If the integer value to be converted exceeds the value which can be represented at the outputs, the error signal output is set. In addition, the output is limited to the upper or lower limit value.

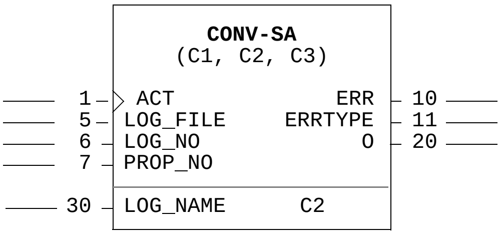


Function diagram

Code Converter, String to Array CONV-SA

Summary

CONV-SA (**CONV**ert **S**tring to **A**rray) converts a string in the data base to an array which is presented on the output of the element.



PC element CONV-SA

Call CONV-SA (C1, C2, C3)

Parameter	Significance	Permissible values
C1	Number of characters in output	1 - 80
C2	Selection of symbolic searching	0 - 1
C3	Number of characters in input LOG__NAME	1 - 28

Terminal Description

No	Name	Type	Description
1	ACT	IB	**ACT**ivate. Dynamic input to activate the element.
5	LOG_FILE	II	**LOG**ical **FILE**. Indicates in which logical file the requested string is found.
6	LOG_NO	II	**LOG**ical **N**umber. Indicates which instance number containing the required string.
7	PROP_NO	II	**PROP**erty **N**umber. Indicates which property contains the required string.
10	ERR	OB	**ERR**or. The output is set to 1 if the specified name is not found in the data base, or if the name is not of data type string.
11	ERRTYPE	OI	**ERR**or **TYPE**. Output to decide type of error.
20	O	OA*C1*	**O**utput. Output for the text.
30	LOG_NAME	IA*C3*	**LOG**ical **NAME**. Only used if LOG_NO=0. LOG_NO=0. Indicates which name contains the required string. Available only if C2=1

Function

When the element detects a positive edge at input ACT, the content of the specified string in the data base is copied to the output of the element as data type array. The required string is addressed with the input terminals, LOG_FIL, LOG_NO and PROP_NO.

If the input terminal LOG_NO=0, the name of the input terminal LOG_NAME is used for a symbolic searching.

The number or characters at the input for logical name is determined by call parameter *C3*. The number of characters in the output is determined by call parameter *C1*. If the specified point in the data base does not exists, or is not of data type string, the output ERR is set to 1.

NOTE

The execution time at positive transition of ACT is considerably greater than the normal case.

Error Supervision

When the element has detected an error, the output ERR is set to 1, and at the same time, the type of error is indicated at output ERRTYPE. The element will indicate following error types:

0. No errors are detected. (ERR is set to 0).

1. Selected logical file not found.

2. Selected logical number not found.

3. Selected property number not found.

4. Selected symbolic name not found.

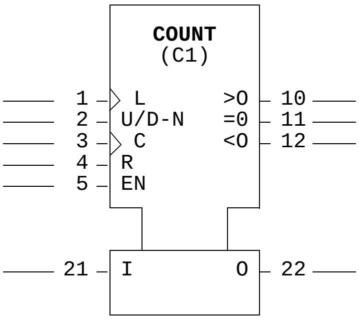
5. Specified point in the data base is not of data type string.

Counter

COUNT

Summary

COUNT (**COUNT**er) is a presettable counter for counting pulses, up or down. The counter also monitors the relation of the counter value to 0.



PC element COUNT

Call COUNT (C1)

Parameter	Significance	Permissible values
C1	Data type	I, IL

Terminal Description

No	Name	Type	Description
1	L	IB	**L**oad. **L**oads the counter with the value at input I.
2	U/D- N	IB	**U**p/**D**own-**N**. Input which determines if the counter is to count up (U/D-N=1) or down (U/D-N=0).
3	C	IB	**C**lock. When this input changes from 0 to 1 the counter counts up or down in accordance with the status of the input U/D-N.
4	R	IB	**R**eset. Input which clears the counter and prevents all further counting or loading.
5	EN	IB	**EN**able. Input which is to be 1 to permit counting or loading. Reset execution is, however, performed independently of EN.
10	>0	OB	Output which is set when the value of the counter is greater than 0.
11	=0	OB	Output which is set when the value of the counter is 0.
12	<0	OB	Output which is set when the value of the counter is less than 0.
21	I	I*C1*	**I**nput. Input for new value when loading.
22	O	O*C1*	**O**utput. Output for counter value.

Function

The counter value increases or decreases by 1 immediately when input C is set. The value increases if U/D-N=1, and decreases if U/D-N=0. The duration of the counter period may not be less than 2 x the cycle time of the program.

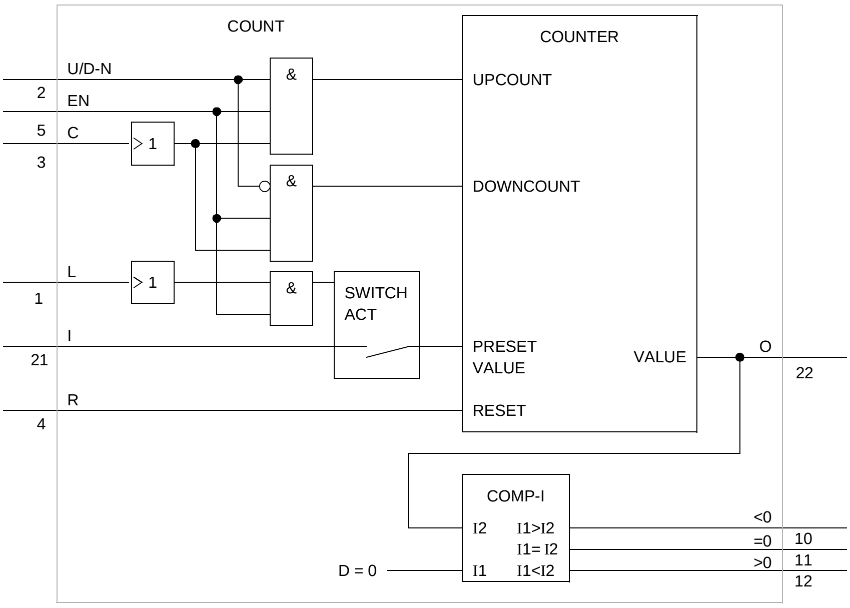
When the input L is set, the counter is loaded with the value at input I. If both input L and input C are set simultaneously, the counter is first loaded after which an up or down count is performed. The input EN must be set for the counter to count or load a new value

Clearing

The input R clears the counter and prevents all further counting or loading. R overrides EN.

Supervision

The status outputs specify the relation of the counter value to zero (>0, =0, <0). When the counter reaches its least or greatest value for the data type, all counting ceases.

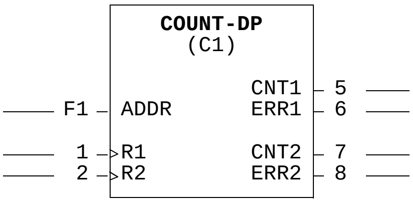


Function diagram

Pulse Counter COUNT-DP

Summary

COUNT-DP (**COUNT**er **- DP**) is used with DSDP 150 for counting pulses at frequencies up to 10 kHz.



PC element COUNT-DP

Call COUNT-DP (C1)

Parameter	Significance	Permissible values
C1	Indicates the group number of counter 1 in the group of two counters used.	1 - 6

Terminal Description

No	Name	Type	Description
F1	ADDR	II	**ADDR**ess. Strapped I/O address on DSDP 150 in decimal code. The board must be strapped within the interval for Misc Board which is address 192 to 239.
1	R1	IB	**R**eset **1**. Dynamic input for clearing counter 1 at change from 0 to 1.
2	R2	IB	**R**eset **2**. Dynamic input for clearing counter 2 at change from 0 to 1.
5	CNT1	OIL	**C**ou**NT 1**. Output for accumulated pulse count for counter 1.
6	ERR1	OB	**ERR**or **1**. Error signal that indicates overflow or addressing error on DSDP 150.
7	CNT2	OIL	**C**ou**NT 2.** Output for accumulated pulse count for counter 2.
8	ERR2	OB	**ERR**or **2**. Error signal that indicates overflow or addressing error on DSDP 150.

Function

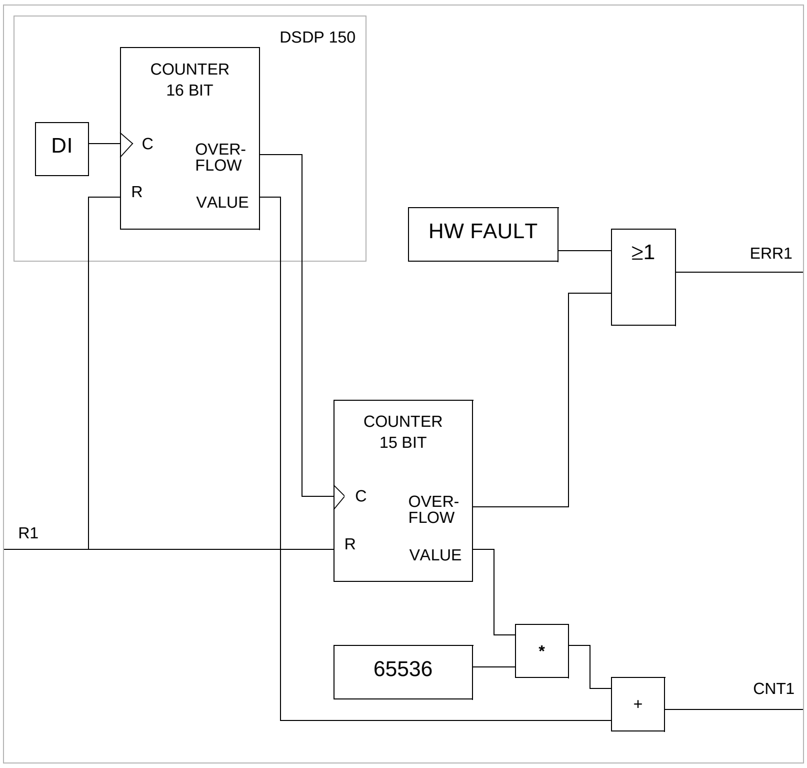
The element handles 2 of the 12 individual counters of DSDP 150. The first counter is controlled by the terminals R1, CNT1 and ERR1. Counter 2 is controlled by the terminals R2, CNT2 and ERR2. Pulses from the process are continuously counted in the hardware counter. The maximum value for the counter is 16 bits+ overflow 131071. At each execution, possible overflows are accumulated in a counter in the PC element. This value, multiplied by 65536 (2^{16}), is added to the current value of the hardware counter CNTx and presented at the output. The max. value of the output signal is 2^{31}. If this value is exceeded, the output ERRx is set and the greatest positive number is presented. The output ERRx is also set to 1 if the COUNT-DP element is unable to reach the DSDP 150.

Clearing

The counter is cleared when Rx goes to 1, after having been 0.

Strapping of DSDP 150

Strap group S1XX, where XX denotes the number of the counter, must be strapped according to the following when used with COUNT-DP: 1–2 and 7–8.

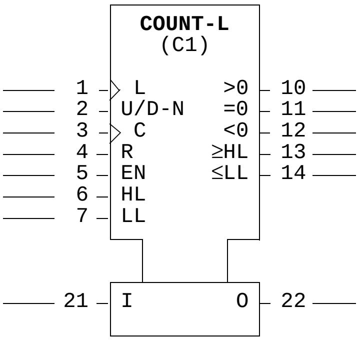


Function diagram, channel

Counter with Limits COUNT-L

Summary

COUNT-L (**COUNT**er - **L**imiter) is a presettable counter for counting up or down. The counter also checks the counter value relative to 0 and optional max and min values.



PC element COUNT-L

Call COUNT-L (C1)

Parameter	Significance	Permissible values
C1	Data type	I, IL

Terminal Description

No	Name	Type	Description
1	L	IB	**L**oad. Loads the counter with the value at input I.
2	U/D- N	IB	**U**p/**D**own-**N**. Input which determines if the counter is to count up (U/D-N=1) or down (U/D-N=0).
3	C	IB	**C**lock. When this input changes from 0 to 1, COUNT-L counts up or down depending on the status at input U/D-N.
4	R	IB	**R**eset. Input which clears the counter and prevents all further counting or loading.
5	EN	IB	**EN**able. Input which is to be 1 for counting or loading to take place. Reset execution takes place irrespective of EN.
6	HL	I*C1*	**H**igh **L**imit. Input for the upper limit.
7	LL	I*C1*	**L**ow **L**imit. Input for the lower limit.
10	>0	OB	Output which is set when the value of the counter is greater than 0.
11	=0	OB	Output which is set when the value of the counter is 0.
12	<0	OB	Output which is set when the value of the counter is less than 0.
13	≥HL	OB	Output which is set when the value of the counter is greater than or equal to the upper limit.
14	≤LL	OB	Output which is set when the value of the counter is less than or equal to the lower limit.
21	I	I*C1*	**I**nput. Input for new value when loading.
22	O	O*C1*	**O**utput. Output for counter value.

Function

The counter value increases or decreases by 1 immediately when input C is set. The value increases if U/D-N=1, and decreases if U/D-N=0. The duration of the counter period may not be less than 2 x the cycle time of the program.

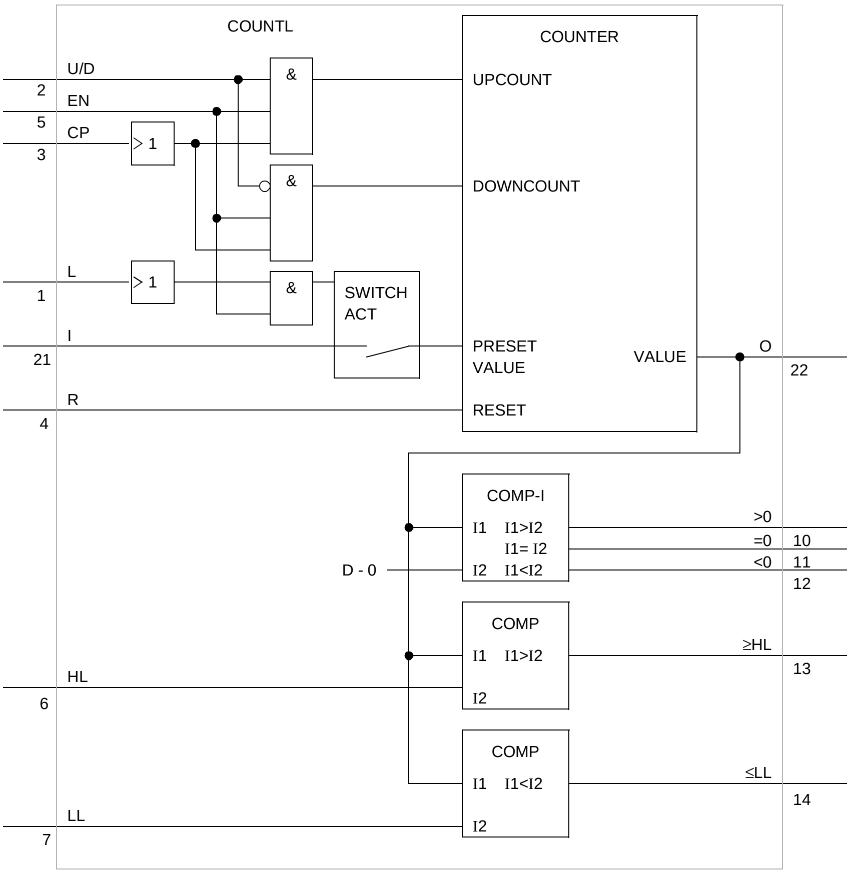
When input L is set, the counter is loaded with the value at input I. If both input L and input C are set simultaneously, the counter is first loaded after which an up or down count is performed. The input EN must be set for the counter to count or load a new value.

Clearing

The input R clears the counter and prevents all further counting or loading. R overrides EN.

Supervision

The status outputs specify the relation of the counter value to zero (>0, =0, <0), and the relation to the upper (≥HL), and lower (≤LL) limits. When the counter reaches its least or greatest value for the data type, all counting ceases.

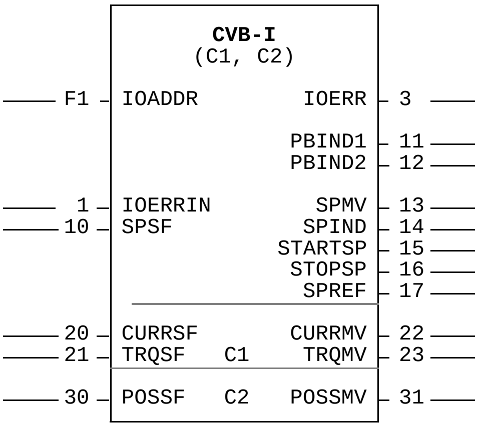


Function diagram

Digital Speed Measurement CVB-I

Summary

The CVB-I (Input from DSDC 111 board) PC element is used to read measured values, indications and error signals from the digital speed measurement and speed control board DSDC 111.



PC element CVB-I

Call CVB-I (C1, C2)

Parameter	Significance	Permissible values
C1	PC element outputs for measured current and torque to be used. Actual current feedback connected to A1 of the DSDC 111 board.	0,1
C2	PC element output for measured position to be used.	0,1

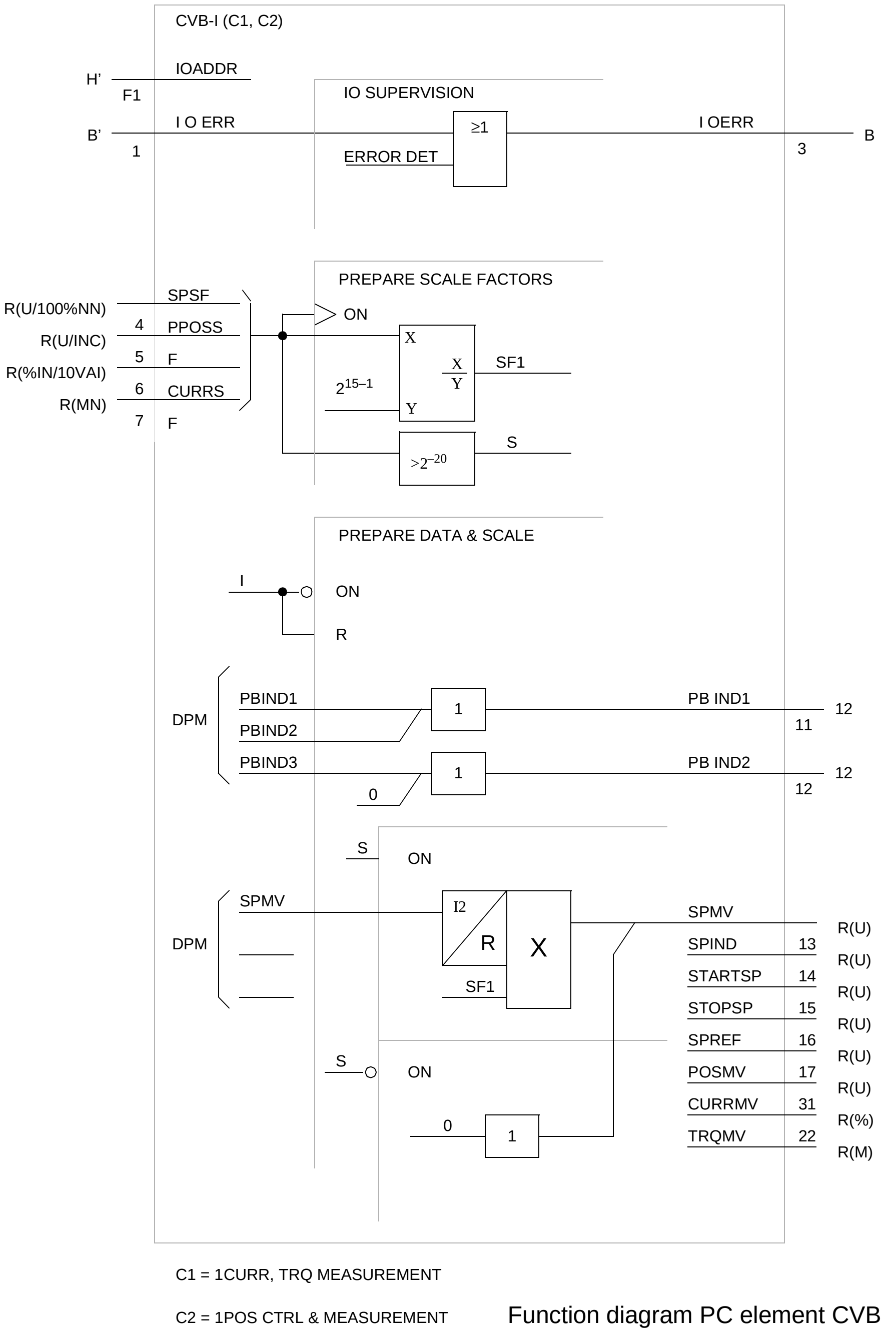
Terminal Description

No	Name	Type	Description
F1	IOADDR	II	Strapped I/O address on DSDC 111 in decimal code. The board must be strapped to an even address within the interval for Misc Board which is address 192 to 239.
1	IOERRIN	IB	**I/OERR**or **IN**put. Should be connected to the IOERR on the related CVB-O element. Set if the DSDC 111 board can not be accessed by the CVB-O PC element.
3	IOERR	OB	**I/OERR**or. Set if the board can not be accessed by the CVB-I PC element. Note that IOERR should be connected to the IOERRIN input signal of the related CVB-O PC element.
10	SPSF	IR	**SP**eed **S**cale **F**actor. Set equal to the value and unit that should be presented for the measured speed outputs, when the rotating speed is equal to the set nominal speed parameter, NNRPM, of the CVB-O PC element.

Terminal Description

No	Name	Type	Description
11	PBIND1	OI	**P**acked **B**oolean **IND**ications. When unpacked in an integer to boolean converting PC element, the indications will appear as follows: 01-RDYRUN. Ready for run 02-RDYREF. Ready for reference 03-NGTN0. Speed exceeding zero speed lim 04-LOCAL. DSDC 111 in local mode 05-TRIPPED. Motor drive tripped 06-ESTOP. Emergency stop 07-NSPLIM. Near max speed 08-NCURRLIM. Near current lim 09-SPREFACK. Speed mode start order ackn. 010-FORW. NGTN0 and forward rotation 011-REV. NGTN0 and reverse rotation 012-POSACK. Pos mode start order ackn. 013-DELINPOS. Delayed in pos ackn. 014-SYNCRDY. Pos synchronization ready 015-LATCH. DSD111 D17 from SCR is set. This DI controls the latch of STARTSP and STOPSP measured values.
12	PBIND2	OI	**P**acked **B**oolean **IND**ications. When unpacked in an integer to boolean converting PC elements, the indications will appear as follows: 01-ONSEQF. Switch on order fault indication 02-STARTSEQF. Start order fault indication 03-MOTOSP. Motor over speed indication 04-NSPFBACK. Pulse tach failure 05-STALL. Motor drive stalled 06-SPREFINLIM. Speed reference exceeding. 100% rated motor speed.
13	SPMV	OR	**SP**eed **M**easured **V**alue. Measured speed in same units as the SPEED SCALE FACTOR. Filter time is determined by the MVFT parameter of the CVB-O PC element.
14	SPIND	OR	**SP**eed **IND**ication. Speed indication in same units as SPEED SCALE FACTOR. Filter time is determined by the INDFT parameter of the CVB-O PC element.
15	STARTSP	OR	**START SP**eed. Actual, memorized speed at the instant the LATCH DI to the DSDC 111 board is set. Value in same units as the SPSF parameter.
16	STOPSP	OR	**STOP SP**eed. Actual, memorized speed at the instant the LATCH DI to the DSDC 111 board is reset. Value in same units as the SPSF parameter.
17	SPREF	OR	**SP**eed **REF**erence. Total sum of all speed references. Value in same units as the SPSF parameter.
20	CURRSF	IR	**CURR**ent **S**cale **F**actor. (*C1*=1). Set equal to the percentage of rated motor current that corresponds to an AI current feedback signal equal to 10V.
21	TRQSF	IR	**T**o**RQ**ue **S**cale **F**actor. (*C1*=1). Set equal to the value and unit that should be presented for the measured torque output when the armature current is equal to 100% of rated motor at base speed.
22	CURRMV	OR	**CURR**ent **M**easured **V**alue. (*C1*=1). Measured current in percentage of rated motor current.
23	TRQMV	OR	**T**o**RQ**ue **M**easured **V**alue. (*C1*=1). Measured torque in same units as the TRQST parameter.
30	POSSF	IR	**POS**ition **S**cale **F**actor. (C2=1). Set equal to the value and unit corresponding to one counted pulse. Note that due to electronic multiplying, the counted number of pulses per pulse tach revolution, is four times rated number of pulses per revolution of a two channel pulse tach.
31	POSMV	OR	**POS**ition **M**easured **V**alue. (C2=1). Measured position in same units as the POSSF parameter.

Function



Function diagram PC element CVB-I

Supervision, IOERRPR Signals

The IOERRIN input should be connected from the CVB-O PC element, writing setpoints and parameters to the same DSDC 111 board.

The IOERR output should be connected to the CVB-O PC element, and to the external, cabled emergency stop circuits of the motor drive.

The logical output IOERR is normally 0 or ”false”. This output indicates a hardware error on DSDC 111. IOERR is set to 1 or ”true” as a result of the following:

- Bus error, the DSDC 111 board is missing or erroneous. System message 131 is sent.
- Access failure to the DSDC 111 board is detected. Software error system message 132 is sent.

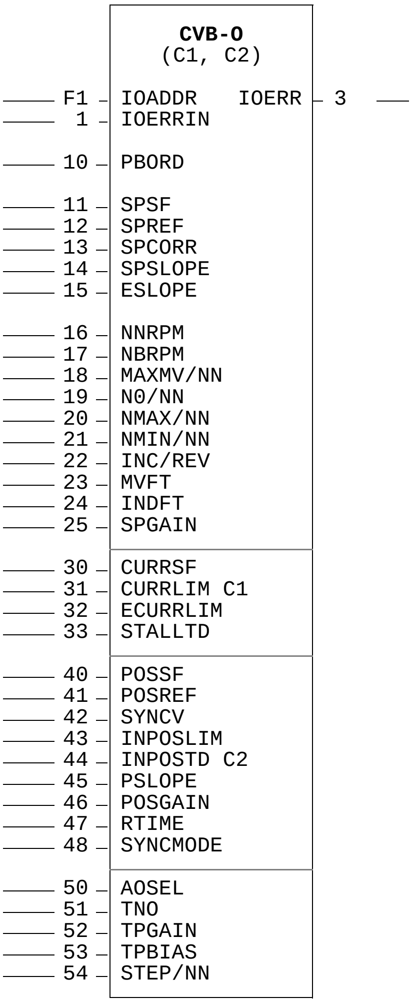
When IOERR is ”true” the PC element will try to access DSDC 111 every 30s, to test if the board can be accessed. IOERR is set to ”false” when the PC program is run in reset mode.

For above type of errors, numerical variables are cleared and boolean signals except IOERR are set to ”false”.

Digital Speed Control **CVB-O**

Summary

The CVB-O (Output to DSDC 111 board) PC element is used to write setpoints, orders and parameters to the digital speed measurement and speed control board DSDC 111.



PC element CVB-O

Call CVB-O (C1, C2)

Parameter	Significance	Permissible values
C1	PC element inputs for current limits and current scale factor to be used. Actual current feedback connected to A1 of the DSDC 111 board.	0 - 1
C2	PC element input for positioning setpoints, orders and parameters to be used.	0 - 1

Terminal Description

No	Name	Type	Description
F1	IOADDR	II	Strapped I/O address on DSDC 111 in decimal code. The board must be strapped to an even within the interval for Misc Board which is address 192 to 239. Strapped **IO ADDR**ess of in decimal code.
1	IOERRIN	IB	**I/OERR**or **IN**put. Should be connected to the IOERR on the related CVB-I element. Set if the DSDC 111 board can not be accessed by the CVB-I element.
3	IOERR	OB	**I/OERR**or. Set if the board can not be accessed by the CVB-0 PC element. Note that IOERR should be connected to the IOERRIN input signal of the related CVB-I PC element.
10	PBORD	II	**P**acked **B**oolean **ORD**ers. When packed in an boolean to integer converting PC element, the orders should be connected as follows: I1 - ON. Switch on order I2 - OFF_N. Negated off order I3 - START. Start order (phase advance) I4 - REFREL. Release speed ref I5 - VARSLOPE. Variable ramp slope I6 - ESTOP_N. Negated emergency stop order I7 - TRIP_N. Negated trip order I8 - (not used) I9 - STARTPOS. Positioning start order I10 - RESYNCRDY. Reset sync. ready I11 - INHESYNC.Inhibit sync. of pos counter. I12 - PGMSYNC. Program controlled sync. order I13 - (not used) I14 - (not used) I15 - (not used
11	SPSF	IR	**SP**eed **S**cale **F**actor. Set equal to the value and unit of the speed reference that should result in a motor speed equal to the set nominal speed parameter, NNRPM.
12	SPREF	IR	**SP**eed **REF**erence. Speed reference in same units as the SPSF parameter. The speed reference level determines the max speed limit for the position control.
13	SPCORR	IR	**SP**eed **CORR**ection. Speed correction reference in same units as the SPSF parameter.
14	SPSLOPE	IR	**SP**eed **SLOPE**. Speed reference linear ramp slope in %/s.
15	ESLOPE	IR	**E**mergency **SLOPE**. Emergency stop linear ramp slope in %/s.
16	NNRPM	IR	Nominal motor speed in RPM.
17	NBRPM	IR	Motor Base speed in RPM.
18	MAXMV/NN	IR	**MAX M**easurable speed **V**alue/**N**ominal motor speed. The speed relation is given in %. (100%<MAXMV/NN<199%).
19	N0/NN	IR	Zero speed **N0/N**ominal motor speed. The speed relation is given in %.
20	NMAX/NN	IR	**MAX** motor speed/**N**ominal motor speed. The speed relation., given in % (>100%), defines the trip level for the overspeed protection.
21	NMIN/NN	IR	**MIN** motor speed/**N**ominal motor speed. The speed relation, given in %, defines together with the STALLTD the trip condition for the motor stall protection.
22	INC/REV	IR	**INC**rements/**REV**olution. Four times the rated number of pulses per revolution of the two channel pulse tach. The four times multiplier is a result of electronic multiplication of the two pulse channels.

Terminal Description

<table>
<tr><th>No</th><th>Name</th><th>Type</th><th>Description</th></tr>
<tr><td>23</td><td>MVFT</td><td>II</td><td>**M**easured **V**alue **F**ilter **T**ime. Filter time, in ms, for the filtering of the speed feedback value used for speed control.</td></tr>
<tr><td>24</td><td>INDFT</td><td>II</td><td>**IND**icated **V**alue **F**ilter **T**ime. Filter time, in ms, for the filtering of the speed feedback value used for speed indication. (>20 ms).</td></tr>
<tr><td>25</td><td>SPGAIN</td><td>IR</td><td>**SP**eed deviation **GAIN**. Gain factor for the speed control deviation. A gain of 10 means that a speed deviation of 10% results in an analog output signal of 10V. This gain factor is used to reduce the influence of offset errors in the analog speed regulator connected to the analog speed error output form the DSDC 111 board.</td></tr>
<tr><td>30</td><td>CURRSF</td><td>IR</td><td>**CURR**ent **S**cale **F**actor. (*C1*=1). Set equal to the percentage of rated motor current that corresponds to an AI current feedback signal equal to 10V.</td></tr>
<tr><td>31</td><td>CURRLIM</td><td>IR</td><td>**CURR**ent **LIM**it. (*C1*=1). Set equal to max current in percentage of rated motor current. Depending on the setting of the AOSEL parameter, the set limit is available on the second AO of the DSDC 111.</td></tr>
<tr><td>32</td><td>ECURRLIM</td><td>IR</td><td>**E**mergency stop **CURR**ent **LIM**it. (1=1). Set equal to max current in percentage of rated motor current. Depending on the setting of the AOSEL parameter, the set limit is available on the second AO of the DSDC 111 board.</td></tr>
<tr><td>33</td><td>STALLTD</td><td>II</td><td>**STALL T**ime **D**elay. (*C1*=1). Set time delay in ms for the motor drive stall protection.</td></tr>
<tr><td>40</td><td>POSSF</td><td>IR</td><td>**POS**ition **S**cale **F**actor. (C2=1). Set equal to the value and unit corresponding to one counted pulse. Note that due to electronic multiplying, the counted number of pulses per pulse tach revolution, is four times the rated number of pulses per revolution of the pulse tach.</td></tr>
<tr><td>41</td><td>POSREF</td><td>IR</td><td>**POS**ition **REF**erence. (C2=1). Position reference in same units as the POSSF parameter.</td></tr>
<tr><td>42</td><td>SYNCV</td><td>IR</td><td>**SYNC**hronization **V**alue. (C2=1). Synchronization counter preset value in same units as the POSSF parameter.</td></tr>
<tr><td>43</td><td>INPOSLIM</td><td>IR</td><td>**IN POS**ition **LIM**it. (C2=1). Limit for IN POSITION indication in same units as the POSSF parameter.</td></tr>
<tr><td>44</td><td>INPOSTD</td><td>II</td><td>**IN POS**ition **T**ime **D**elay. (C2=1). Time delay in ms for the IN POSITION indication.</td></tr>
<tr><td>45</td><td>PSLOPE</td><td>IR</td><td>**POS**itioning **SLOPE**. (C2=1). Positioning linear ramp slope in %/s.</td></tr>
<tr><td>46</td><td>POSGAIN</td><td>IR</td><td>**POS**ition control **GAIN**. (C2=1). Reduces the calculated slopes of the parabol function, converting a position deviation to a speed reference. (<1.0).</td></tr>
<tr><td>47</td><td>RTIME</td><td>IR</td><td>**R**aise **TIME**. (C2=1). Set as the speed control response time in s. Determines the slope of the parabol function for small position errors.</td></tr>
<tr><td>48</td><td>SYNCMODE</td><td>II</td><td>**SYNC**hronization **MODE**. (C2=1). Determines the condition for the position counter to be synchronized and preset with the set SYNCValue.
<table>
<tr><th>Sync DI
Lead edge</th><th>Sync DI
Trail edge</th><th>Other condition</th></tr>
<tr><td>0</td><td>1</td><td></td></tr>
<tr><td>2</td><td>3</td><td>Strobe</td></tr>
<tr><td>4</td><td>5</td><td>Strobe &Up direction</td></tr>
<tr><td>6</td><td>7</td><td>Strobe & Down direction</td></tr>
<tr><td>8</td><td>9</td><td>Up direction</td></tr>
<tr><td>10</td><td>11</td><td>Down direction</td></tr>
<tr><td></td><td>12</td><td>Strobe</td></tr>
<tr><td></td><td>14</td><td>Strobe & Up direction</td></tr>
</table>
</td></tr>
</table>

Terminal Description

<table>
<tr><th>No</th><th>Name</th><th>Type</th><th>Description</th></tr>
<tr><td>50</td><td>AOSEL</td><td>II</td><td>AO SELection. The parameter value 0-3 determines what signal is guided to the second AO of the DSDC 111 board.
<table>
<tr><th>AOSEL</th><th>Signal</th></tr>
<tr><td>0- 2</td><td>Measured speed. 10V=nominal speed.</td></tr>
<tr><td>1</td><td>Current limit. 10V=the relation to rated motor current given by the CURRSF parameter.</td></tr>
<tr><td>3</td><td>Measured torque. 10V=the relation to rated motor torque given by the CURRSF parameter.</td></tr>
</table></td></tr>
<tr><td>51</td><td>TPNO</td><td>II</td><td>Test Point Number. (1-32). Guides various testpoint signals to the test AO available on the front of the DSDC 111 board. When given a value over 128, the value is interpreted by the board to be a memory address.</td></tr>
<tr><td>52</td><td>TPGAIN</td><td>II</td><td>Test Point GAIN. (0-128). Amplifies the testpoint value by 2**TPGAIN.</td></tr>
<tr><td>53</td><td>TPBIAS</td><td>IR</td><td>Test Point BIAS. (0-100%). Determines the bias level of the testpoint signal. The bias value is subtracted from the value before the value is amplified with the set TPGAIN.</td></tr>
<tr><td>54</td><td>STEP/NN</td><td>IR</td><td>Speed STEP in relation to Nominal speed. Given in % of nominal speed. The speed step is activated by the DDS push button, located on the front of the DSDC 111 board.</td></tr>
</table>

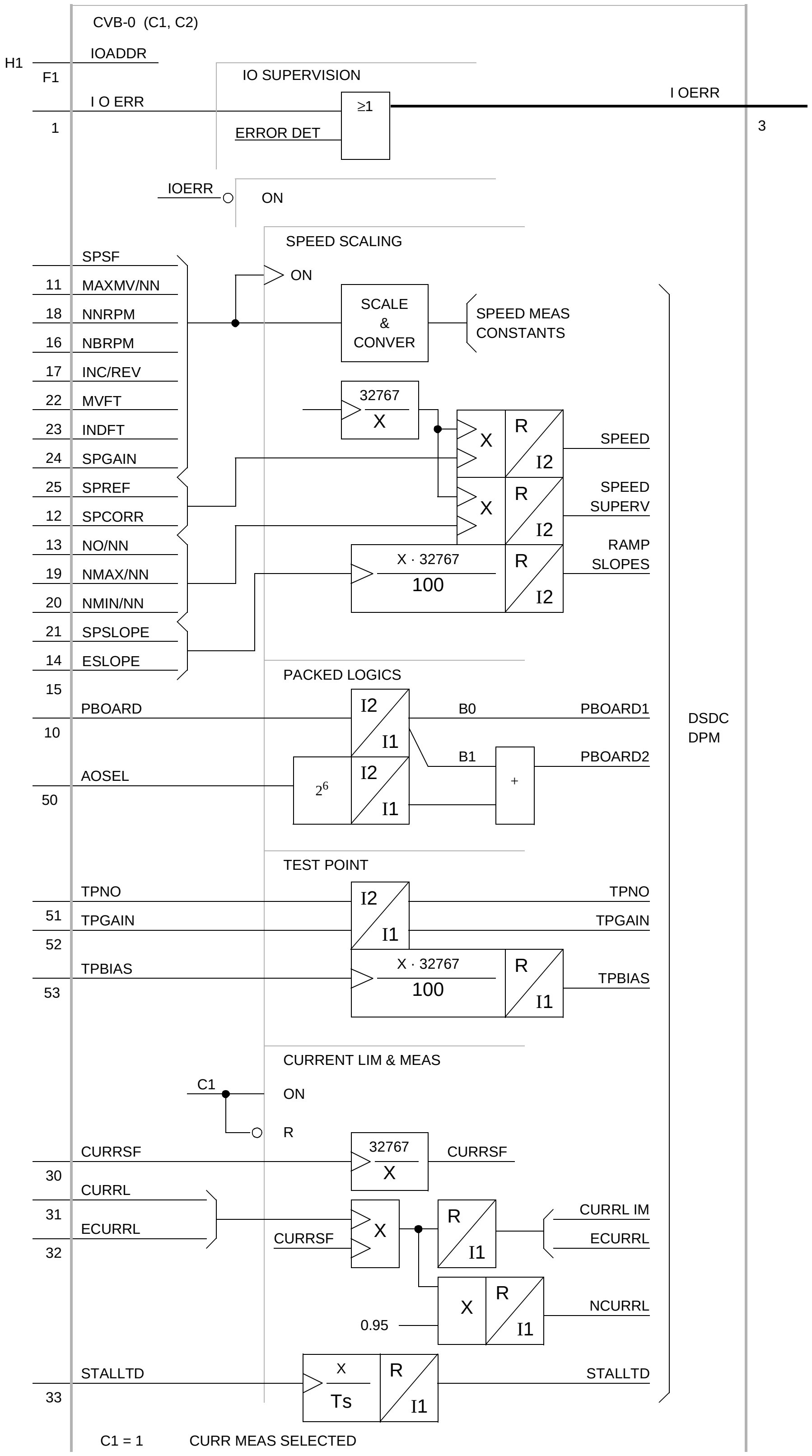
Supervision, IOERRPR Signals

The IOERRIN input should be connected from the CVB-I PC element, reading values and indications from the same DSDC 111 board. The IOERR output should be connected to the CVB-I PC element and to the external, cabled emergency stop circuits of the motor drive. The logical output IOERR is normally 0 or ”false”. This output indicates a hardware error on DSDC 111. IOERR is set to 1 or ”true” as a result of the following:

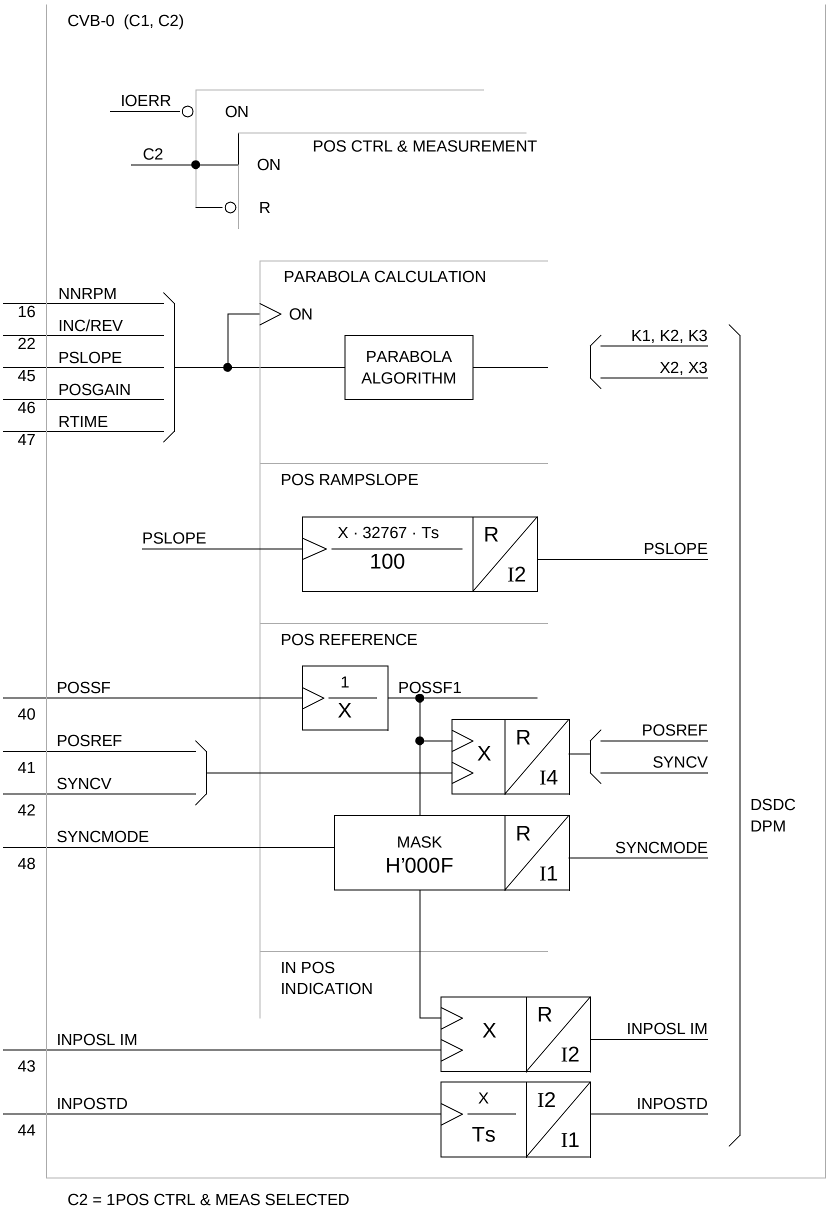
- Bus error, the DSDC 111 board is missing or erroneous. System message 131 is sent.
- Access failure to the DSDC 111 board is detected. Software error system message 132 is sent.

When IOERR is ”true” the PC element will try to access DSDC 111 every 30s to test if the board can be accessed. IOERR is set to ”false” when the PC program is run in reset mode. For above type of errors, numerical variables are cleared and boolean signals except IOERR are set to ”false”.

Function



Function diagram
PC element CVB-O

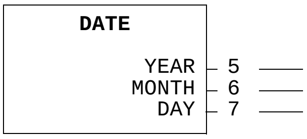


Function diagram PC element CVB-O, continued

Date Reading DATE

Summary

DATE is used to get the current date.



PC element DATE

Call DATE

Terminal Description

No	Name	Type	Description
5	YEAR	OI	Output for Year (≥1980).
6	MONTH	OI	Output for Month (1-12).
7	DAY	OI	Output for Day (1-31).

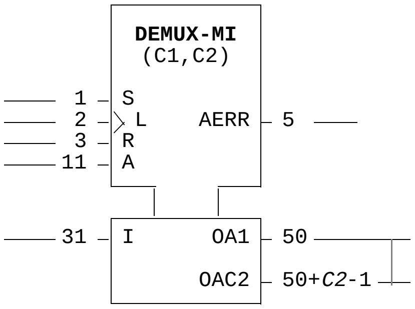
Function

The current date is given at the three outputs YEAR, MONTH and DAY. These outputs may, by using external arithmetic and comparator elements, be used to activate a function on a given day. The element compensates for leap years.

Demultiplexer with Memory DEMUX-MI

Summary

DEMUX-MI (**DEMU**ltiple**X**er **-** with **M**emory and **I**nteger address) is used as a demultiplexer with memory function. DEMUX-MI has an optional number of outputs. The data type can be integers, real numbers, Boolean variables or time.



PC element DEMUX-MI

Call DEMUX-MI (C1, C2)

Parameter	Significance	Permissible values
C1	Data type	I, IL, R, B, T, TR
C2	Number of outputs	1 - 19

Terminal Description

No	Name	Type	Description
1	S	IB	**S**et. Input for loading of new values each time the element is executed. When the input S is reset (to 0), the value most recently loaded will remain at the outputs.
2	L	IB	**L**oad. Dynamic input for loading data from input I to the output addressed by the input A.
3	R	IB	**R**eset. Input for resetting the outputs. This input overrides the inputs S and L.
5	AERR	OB	**A**ddress **ERR**or. Output which is set to 1 when the address value is greater than the number of outputs, or negative.
11	A	II	**A**ddress. Input for address value which indicates at which output data from input I is to be stored. If the value is 0, the value 0 is stored at all outputs.
31	I	I*C1*	**I**nput. Data input.
50	OA1	O*C1*	**O**utput **A**ddress **1**. The first output from the demultiplexer.
51	OA2	O*C1*	**O**utput **A**ddress **2**. The second output from the demultiplexer.
: 50+*C2*-1	OAC2	O*C1*	**O**utput **A**ddress ***C2***. The last output from the demultiplexer.

]

Function, Addressing

At which output (OA1 - OA*C2*) the data value at input I is to be stored is specified at input A with an address (integer 1 - *C2*). If the address A is 0, the value 0 is stored at all outputs.

Loading

The value is stored at the instant that the input L is set (to 1). If input S is set, new data is loaded each time the element is executed. When S is reset after having been set, the data most recently loaded remains until the element is executed with one of the inputs S, L or R set. The input S overrides the input L, that is, when S is set, L has no effect.

Clearing

Setting the input R clears the data outputs and prevents further storage of data.

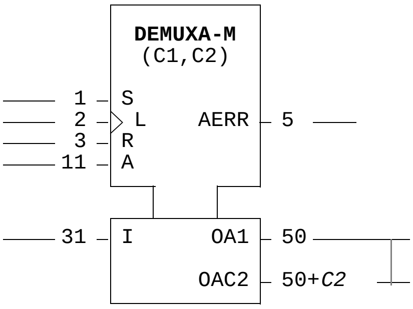
Supervision

The address A is monitored and if its value is greater than the number of outputs or is negative, the error signal output AERR is set. The data value 0 is then stored at all outputs.

Array Demultiplexer DEMUXA-M

Summary

DEMUXA-M (**DEMU**ltiple**X**er **A**rray **-** with **M**emory and integer address) is used as a demultiplexer with memory function.



PC element DEMUXA-M

Call DEMUXA-M (C1, C2)

Parameter	Significance	Permissible values
C1	Number of the characters for the outputs	1 - 80
C2	Number of outputs	1 - 19

Terminal Description

No	Name	Type	Description
1	S	IB	**S**et. Input for loading of new values each time the element is executed. When input S is reset (to 0), the value most recently loaded will remain at the outputs.
2	L	IB	**L**oad. Dynamic input for loading of data from the input I to the output addressed by input A.
3	R	IB	**R**eset. Input for resetting the outputs. This input overrides the inputs S and L.
5	AERR	OB	**A**ddress **ERR**or. Output which is set to 1 when the address value is greater than the number of outputs, or negative.
11	A	II	**A**ddress. Input for address value which indicates at which output data from the input I is to be stored. If the value is 0, the value 0 is stored at all outputs.
31	I	IA*C1*	**I**nput. Data input.
50	OA1	OA*C1*	**O**utput **A**ddress **1**. The first output from the demultiplexer.
51	OA2	OA*C1*	**O**utput **A**ddress **2**. The second output from the demultiplexer.
: 50+*C2*- 1	OA*C2*	OA*C1*	**O**utput **A**ddress ***C2***. The last output from the demultiplexer.

Function, Addressing

At which output (OA1 - OA*C2*) the data value at input I is to be stored is specified at input A with an address (integer 1 - *C2*). If the address A is 0, the value 0 is stored at all outputs.

Loading

The value is stored at the instant that the input L is set (to 1). If input S is set, new data is loaded each time the element is executed. When S is reset after having been set, the data most recently loaded remains until the element is executed with one of the inputs S, L or R set. The input S overrides the input L, that is, when S is set, L has no effect.

Clearing

Setting the input R clears the data outputs and prevents further storage of data, that is, an empty string is stored at all outputs.

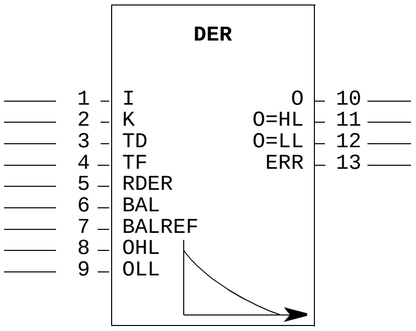
Supervision

The address A is monitored, and if its value is greater than the number of outputs, or negative, the error signal output AERR is set. An empty string is then stored at all outputs.

Derivator DER

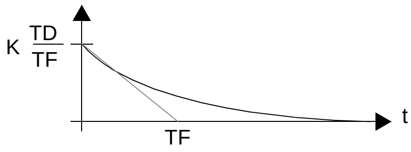
Summary

DER (**DER**ivator) is used to give derivation effect.
The derivation effect can be limited with the filter function, which serves as a low pass filter. The output signal can be limited with limit values specified at special inputs.
The balancing function permits the output signal to follow an external reference and permits a bumpless return to the normal function. All transfers from static states are bumpless.



PC element DER

Call DER



Step response

Terminal Description

No	Name	Type	Description
1	I	IR	**I**nput. Input for actual value.
2	K	IR	Input for setting gain.
3	TD	ITR	**T**ime **D**erivation. Input for time constant for derivation.
4	TF	ITR	**T**ime **F**ilter. Input for filter time constant.
5	RDER	IB	**R**eset **DER**ivator. Input for clearing derivator.
6	BAL	IB	**BAL**ance. Input for activation of following.
7	BALREF	IR	**BAL**ance **REF**erence. Input for reference value when following.
8	OHL	IR	**O**utput **H**igh **L**imit. Input for upper limit value.
9	OLL	IR	**O**utput **L**ow **L**imit. Input for lower limit value.
10	O	OR	**O**utput. Output signal.
11	O= HL	OB	**O**utput=**H**igh **L**imit. Output which is set to 1 if the output is limited to the upper limit value.
12	O= LL	OB	**O**utput=**L**ow **L**imit. Output which is set to 1 if the output is limited to the lower limit value.
13	ERR	OB	**ERR**or. Output which is set to 1 if OHL is less than OLL.

Function

The step response in the time plane for a derivator is:

$O(t)=K\ (TD/TF)e^{-t/TF} \times I(t)$

where I(t) specifies the magnitude of the step.

The transfer function for a DER function is:

$G(s)=K(s \times TD)/(1+s \times TF)$

This has been implemented in the DER element as a recursive algorithm. The design of the algorithm is such that normal functioning is maintained even during limiting. This ensures a controlled return to a dynamic state.

Gain, Derivation Time, Filter Time and Sampling Time

Certain constants are precalculated to make the execution time of the element as short as possible. The results are stored internally in the element. These constants are recalculated if TD, TF or K are changed by more than 1/128 of their previous value or if the sampling time TS is changed. When recalculating a test is performed to check whether TD and TF≥2 x TS.
If not, TD and/or TF is set equal to 2 x TS.

Clearing of the Derivator

Both output O(t) and the internal state of the output are cleared when RDER goes to 1.

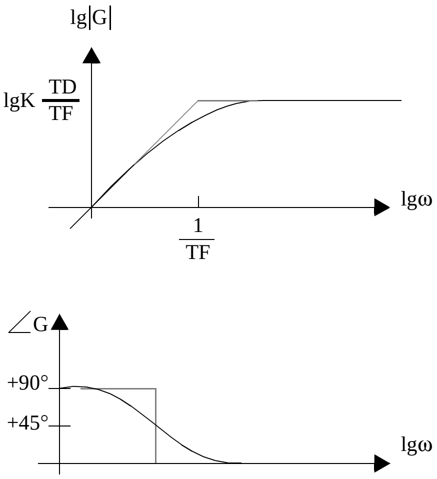
Following

If BAL is set to 1, the derivator immediately goes into following and the output O is set to the value of the input BALREF.
If the value at BALREF exceeds the output signal limits, the output is set to the limit value concerned. Return to dynamic state is bumpless.

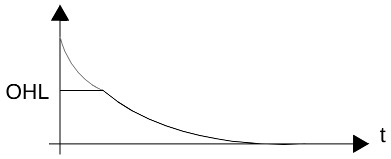
Limitation Function

The limitation function limits the output signal to the values at the inputs OHL for upper limit, and OLL for lower limit. If the actual value exceeds the upper limit, the output O=HL is set to 1. If it falls below the lower limit, the output O=LL is set to 1.

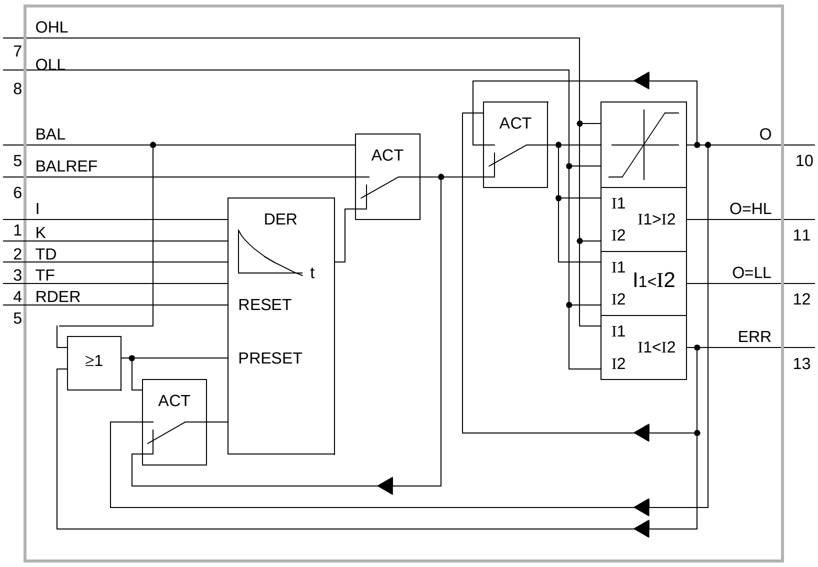
The element checks that the upper limit value OHL is greater than the lower limit value OLL. If not, the output ERR is set to 1. While the error status persists, the outputs O=HL, O=LL and O retain the values they had in the sample before, in which the error occurred. After an error, the return to a dynamic state is bumpless, in the same way as in the case above.



Bode diagram



Function in a limiting state

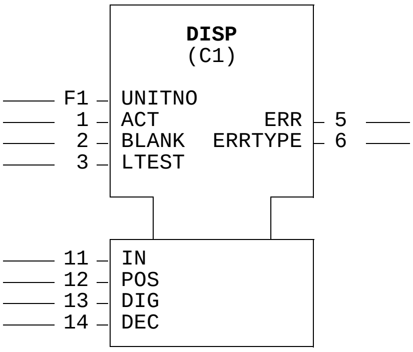


Function diagram

Numeric Display DISP

Summary

DISP (**DISP**lay) is used to show values on numeric display DSHM 110 or DSHM 120. It can handle several data types and be controlled to affect only some of the digits on the numeric display.



PC element DISP

Call DISP (C1)

Parameter	Significance	Permissible values
C1	Data type of the value to be loaded	B, I, IL, R, T, TR

Terminal Description

No	Name	Type	Description
F1	UNITNO	FI	**UNIT N**umber. Address for MasterView 100 unit. The hundreds and thousands indicate the number of the MasterView 100 bus, the units and tens indicate the number of the node.
1	ACT	IB	**ACT**ivate. Input for activated presentation.
2	BLANK	IB	**BLANK**ing. Blanking of display.
3	LTEST	IB	**L**amp **TEST**. Input for activation of lamp test.
5	ERR	OB	**ERR**or. Output for indication of error.
6	ERRTYPE	OI	**ERR**or **TYPE**. Output to determine type of error.
11	IN	I*C1*	**IN**put. Data input.
12	POS	II	**POS**ition. Input for position of first displayed digit to be updated by this element.
13	DIG	II	**DIG**its. The number digits to be updated.
14	DEC	II	**DEC**imals. Number of decimal places.

Function

DISP is used to show values on the numeric display DSHM 110 or DSHM 120. The values which can be shown can be of one of several available data types, as determined by the call parameter *C1*. The element can control the complete numeric display or a part of it. A numeric display can be divided into several parts, each controlled from its own DISP element. The part of the numeric display, that is to be controlled from the element, is given at the inputs POS an DIG:

POS gives the figure which is the first digit on the display to be updated from the element. The numeric display has 7 digits numbered 1 - 7 from the left.

DIG indicates how many digits are controlled from the element.

The location of the decimal point is controlled from input DEC.

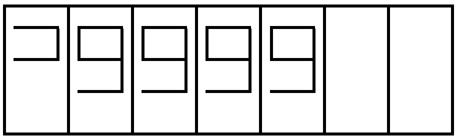
If the element shows Boolean values, DEC is not read and no decimal point is included in the result. With real numbers or real times, DEC specifies how many decimal digits are to be shown. If the data type is integer, DEC indicates how many of the digits in the number are to be shown to the right of the decimal point. When showing time, DIG specifies the number of digits to be present. If DEC = 0, the figures are counted from the left and if DEC is not equal to 0, the figures are counted from the right with the number of decimal places given by DEC for seconds. Example:

DIG	DEC	Number	Shown as
4	0	12:34:56.1234	12.34
6	0	12:34:56.1234	12.34.56
4	3	12:34:56.1234	6.123
6	2	12:34:56.1234	34.56.12

Format

The value presented is read from input IN. Boolean variables are presented as 0 or 1. Leading zeros are shown as blank digits. For numbers with a decimal point, leading zeros up to, but not including the digit before the decimal point are blanked.
Ex: 0.0123.

For data types which can assume negative values, the minus sign is shown to the immediate left of the number when appropriate. Real numbers and real time are always shown in decimal form. Exponents cannot be shown. Real numbers are rounded. Numbers too large to be shown with the number of digits that may be used are indicated with a special character corresponding to > in the first character position, and nines in the remaining positions. The number of fines shown depends on the values at POS and DIG. See the figure below.



Indication of out-of-range number
POS=1
DIG=5

Time Multiplexing of a Numeric Display

The same part of a numeric display can be controlled from several DISP elements at different times. For example, a display may present the values entered during reading-in and process values otherwise.

Activation, Lamp Test and Blanking of Display

Three inputs are used for controlling the element functions ACT, BLANK and LTEST. If ACT is set to 1, the element issues values on the numeric display. If ACT is set to 0, nothing is written to the display and can then be updated by another element. BLANK has no effect on the element function if ACT is set to 0. If LTEST is set to 1, all segments illuminate irrespective of the status of BLANK and ACT. If ACT = 1 and LTEST = 0, all controlled characters can be blanked by setting BLANK to 1.
The display is then updated with data which corresponds to blanked characters. When BLANK returns to 0, data from the input is presented on the display. By connecting BLANK to an oscillator, the display can be made to flash, for example, to indicate an alarm.

Initialization

At power on, all digit positions on the display are blanked by all elements connected to it. This is to ensure that the positions, not controlled by an element, are blanked.

Error Supervision

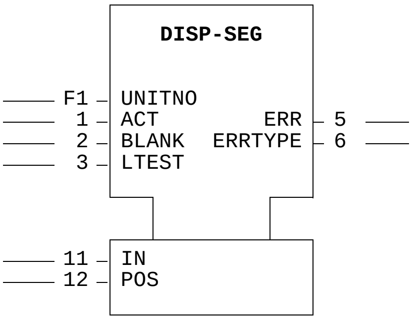
When the element indicates an error, the error output ERR is set to 1, and the type of error is indicated at the same time at the ERRTYPE output. The element can indicate the following error types:

0=The element has not detected an error (ERR=0)
1=Controller board to specified MasterView 100 bus missing
2=Controller board to specified MasterView 100 bus faulty
3=The addressed MasterView 100 unit is missing or faulty
11=Illegal value at POS input
12=Illegal value at DIG input
13=Illegal value at DEC input
15=Illegal value at function parameter UNITNO.

Special Character Display DISP-SEG

Summary

DISP-SEG (**DISP**lay-**SEG**ment) is used to show optional special characters on numeric display DSHM 110 or DSHM 120.



PC element DISP SEG

Call DISP-SEG

Terminal Description

No	Name	Type	Description
F1	UNITNO	FI	**UNIT N**umber. Address of MasterView 100 unit. The hundreds and thousands indicate the number of the MasterView 100 bus, the units and tens indicate the node number.
1	ACT	IB	**ACT**ivate. Input for activated presentation.
2	BLANK	IB	**BLANK**ing of display.
3	LTEST	IB	**L**amp **TEST**. Input for activation of lamp test.
5	ERR	OB	**ERR**or. Output for indication of error.
6	ERRTYPE	OI	**ERR**or **TYPE**. Output to determine type of error.
11	IN	II	**IN**put. Data input.
12	POS	II	**POS**ition. Input for position of first displayed digit to be updated by this element.

Function

The element DISP-SEG is used to present special characters on numeric display DSHM 110 or DSHM 120. The characters can be generated freely by the user, as all character segments can be independently lit or blanked. Each DISP-SEG element interacts with only one character on a numeric display. A physical numeric display must thus be controlled from several DISPSEG or DISP elements.

It is also possible for several DISP-SEG or DISP elements to control the same part of the display, but at different times. A numeric display can, for example, show values entered during the entry process and other values, or special characters, the rest of the time. The POS input is used to determine where on the display the character is to be updated. The characters are numbered 1 - 7 from the left.

Activation, Lamp Test and Blanking of Display

Three inputs are used for controlling the element functions ACT, BLANK and LTEST. If ACT is set to 1, the element activates presentation of the character on the numeric display. If ACT is set to 0, nothing is written to the display, and it can then be updated by another element.

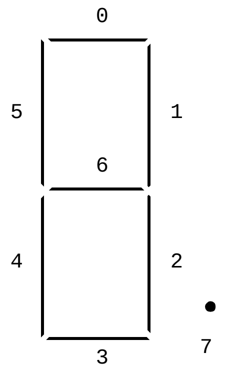
BLANK has no effect on the element function if ACT is set to 0. If LTEST is set to 1, all segments light up irrespective of the status of BLANK and ACT. If ACT = 1 and LTEST = 0, the controlled characters can be blanked by setting BLANK to 1. The display is then updated with data which corresponds to a blanked character. When BLANK returns to 0, data from the input is presented on the display. By connecting BLANK to an oscillator, the display can be made to flash, for example to indicate an alarm.

Format

Each character is made up of segments. Each segment has a value; see the figure on the right hand side below. A special character consists of a combination of segments. This combination is obtained by giving the sum of the segment values as data at the IN input. Ex: The letter d is formed by the segments 1, 2, 3, 4 and 6. The values are summarized (2+4+8+16+64=94), and this value is applied to the IN input.

Segment	Value
0	1
1	2
2	4
3	8
4	16
5	32
6	64
7	128

Segment value



7-segment numbering

Initialization

At power on, all digit positions on the display are blanked by all elements connected to it. This is to ensure that the positions, not controlled by an element, are blanked.

Error Supervision

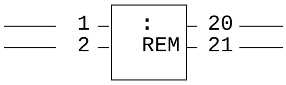
When the element indicates an error, the error output ERR is set to 1, and the type of error is indicated at the same time at the ERRTYPE output. The element can indicate the following error types:

0=The element has not detected an error (ERR=0)
1=Controller board to specified MasterView 100 bus missing
2=Controller board to specified MasterView 100 bus faulty
3=The addressed MasterView 100 unit is missing or faulty
11=Illegal values at POS input
15=Illegal value at function parameter UNITNO.

Divider DIV

Summary

DIV is used for division of two integers or real numbers. When dividing integers, the quotient is obtained with the remainder at a separate input.



PC element DIV

Call DIV (C1)

Parameter	Significance	Permissible values
C1	Data type	I, IL, R

Terminal Description

No	Name	Type	Description
1	-	I*C1*	Input for dividend.
2	-	I*C1*	Input for divisor.
20	-	O*C1*	Output for quotient.
21	REM	O*C1*	**REM**ainder. Output for remainder. Applies only with division of integers.

Function

The value at input 1 is divided by the value at input 2. The quotient is stored at output 20. When dividing integers, the remainder is stored at the output REM.

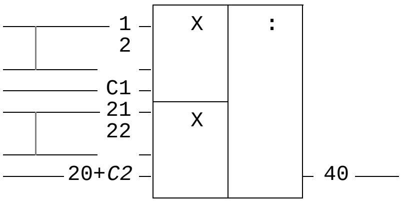
Overflow

If the maximum positive or negative values are exceeded, the output is limited to the highest or lowest representable value for the data type.

Divider with Multipled Inputs DIV-MR

Summary

DIV-MR (**DIV**ider **-** with **M**ultipliers **R**eal numbers) is used for division of two products of real numbers.



PC element DIV-MR

Call DIV-MR (C1, C2)

Parameter	Significance	Permissible values
C1	Number of inputs in the first multiplication element.	1 - 19
C2	Number of inputs in the second multiplication element.	1 - 19

Terminal Description

No	Name	Type	Description
1	-	IR	Factor for dividend.
2	-	IR	Factor for dividend.
: *C1*	-	IR	Factor for dividend.
21	-	IR	Factor for divisor.
22	-	IR	Factor for divisor.
: 20+*C2*	-	IR	Factor for divisor.
40	-	OR	Output for quotient.

Function

The product of the real values at the inputs 1 - *C1* is divided by the product of the real values at the inputs 21 - 20+C2. The quotient is stored at the output 40.

Overflow

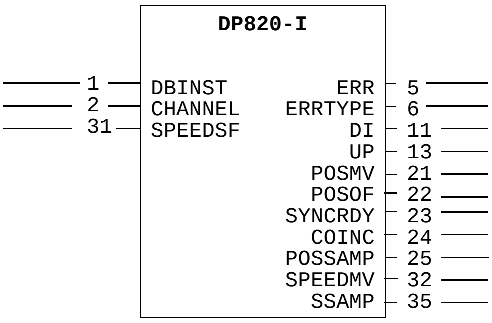
If the maximum positive or negative real values are exceeded, the output is limited to the highest or lowest representable value respectively.

Pulse Counter DP820, Input DP820-I

Summary

Dynamic data (measured values) is read from one channel of the DP820 pulse counter S800 I/O module via Advant Fieldbus 100 by this element.

The module is configured by means of DP820 data base element.



Call DP820-I

Terminal Description

Number	Name	Type	Description
1	DBINST	IIL	**D**ata **B**ase **INST**ance. A reference to the actual DP820 data base element.
2	CHANNEL	II	**CHANNEL** number 1 or 2 to be used.
5	ERR	OB	Output set when an **ERR**or is detected.
6	ERRTYPE	OI	**ERR**or **TYPE**
11	DI	OB	Current state of the **D**igital **I**nput signal.
13	UP	OB	**UP** direction
21	POSMV	OIL	**POS**ition **M**easure **V**alue holds the actual or, when POSSAMP is true, the frozen value of the position counter. Range: -268435456.........+268435455.
22	POSOF	OB	**POS**ition **O**ver**F**low gives a pulse when POSMV exceeds its range and wraps around.
23	SYNCRDY	OB	**SYNC**hronization **R**ea**DY** gives a pulse when Synchronization has occurred.
24	COINC	OB	**COINC**idence gives a pulse when coincidence has occurred.
25	POSSAMP	OB	**POS**ition **SAMP**led value gives a pulse when POSMV is frozen.
31	SPEEDSF	IR	**SPEED S**cale **F**actor scales SPEEDMV from Hz to engineering units. $SPEEDMV = \frac{frequency}{SPEEDSF}$ frequency in Hz
32	SPEEDMV	OR	**SPEED M**easure **V**alue holds the actual or, when SSAMP is true, the frozen frequency value. See also SPEEDSF above.
35	SSAMP	OB	**S**PEED **SAMP**le gives a pulse when SPEEDMV is frozen.

Function

DP820 is a two-channel pulse counter module in the S800 I/O series. Each channel can be used for independent pulse count/length measurement and frequency/speed measurement. The module is placed in a S800 I/O station connected to the Controller via Advant Fieldbus 100.

Configuration of module DP820 is handled via the data base element DP820.

Dynamic data is channeled through these PC elements:

- DP820-O for set points
- DP820-I for measured values

DP820 module offer two main functions:

- Pulse counter measurement. A 29 bits bidirectional counter with:
  - Coincidence detection controlling one digital output signal.
  - Freezing of counter value depending on different conditions.
  - Synchronization of counter depending on different conditions.
- Frequency/Speed measurement up to 1.5 MHz with:
  - Freezing of value depending on different conditions.
  - Selectable measure time within 1 up to 2700 milliseconds.
  - Built-in scaling of frequency value to engineering units.

Transducers with quadrature encoded signals or with up/down pulse signals can be connected to the module.

Speed measurement

The element calculates its speed measure value on its output SPEEDMV terminal like this:

$$SPEEDMV = \frac{frequency}{SPEEDSF} \qquad \text{frequency in Hz}$$

where SPEEDSF is its Speed Scale Factor input terminal value.

If SPEEDSF=0 no scaling is performed; that is SPEEDSF = 1 is assumed.

In case of ERROR the SPEEDMV is frozen, but in case of “Transducer power error” the value is forced to zero.

ERROR HANDLING

The following errors may appear on the ERRTYPE output terminal:

0 OKAY, no error.
1 Data Base element not found
2 Data base element not implemented
3 Data base element not in service
4 Illegal Data base element TYPE
5 Data Base error
6 Transducer power error
7 Access error (including bus error)
9 Invalid channel number
11 Module error

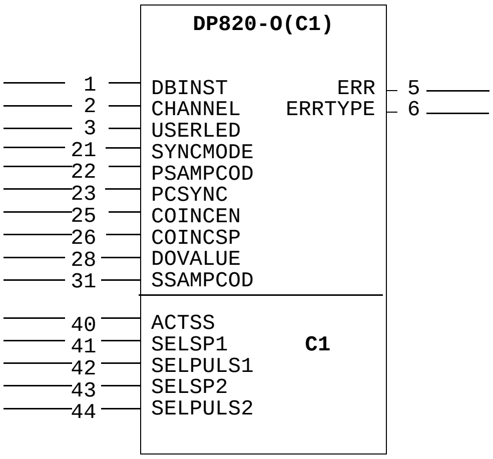
Pulse Counter DP820, Output

DP820-O

Summary

This element is used for sending set points for one channel of the DP820 pulse counter module on Advant Fieldbus 100.

The module is configured by means of DP820 data base element.



Call DP820-O (C1)

Call Parameter

Parameter	Significance	Permissible values	Default values
C1	Selection of Simultaneous Sampling function (Only valid for first channel)	0............1	0

Terminal Description

Number	Name	Type	Description
1	DBINST	IIL	**D**ata **B**ase **INST**ance, a reference to the actual DP820 data base element.
2	CHANNEL	II	**CHANNEL** number 1 or 2 to be used
3	USERLED	IB	Controls the **USER** defined **L**ight **E**mitting **D**iode at the front of the module.
5	ERR	OB	**ERR**or output set when an error is detected.
6	ERRTYPE	OI	**ERR**or **TYPE**

Number	Name	Type	Description
21	SYNCMODE	II	**SYNC MODE.** Choice of synchronization for position counter: 0: Sync inhibit 1: Sync on PCSYNC 2: Sync on strobe and UP direction 3: Sync on strobe and DOWN direction 4: Sync on DI positive transition and UP direction 5: Sync on DI positive transition and DOWN direction 6: Sync on DI negative transition and UP direction 7: Sync on DI negative transition and DOWN direction 8: Sync on positive transition of DI followed by a positive transition of STROBE and UP direction 9: Sync on positive transition of DI followed by a positive transition of STROBE and DOWN direction 10: Sync on negative transition of DI followed by a positive transition of STROBE and UP direction 11: Sync on negative transition of DI followed by a positive transition of STROBE and DOWN direction 12: Sync on COINCidence 13: Sync on COINCidence on the other channel
22	PSAMPCOD	II	**P**osition **SAMP**ling **COD**e 0: Sampling for positive transition of DI 1: Sampling for negative transition of DI 2: Sampling when coincidence on own channel occurs. 3: Sampling when order is given from the simultaneous sampling function. 4: Sampling when coincidence on the other channel occurs.
23	PCSYNC	IB	**P**osition **C**ounter **SYNC**hronization at positive transition if "SYNCMODE"=1. The pulse must be long enough. Use SYNCRDY as an acknowledgment.
25	COINCEN	IB	**COINC**idence detection **EN**abling. If COINC=0 the value of DOVALUE is forced to the digital output.
26	COINCSP	IIL	**COINC**idence **S**et **P**oint. Comparing value for coincidence evaluation. Range: -268435456.........+268435455
28	DOVALUE	IB	Controls the **D**igital **O**utput **VALUE** according to the data base element terminal DOMODEx
31	SSAMPCOD	II	**S**peed **SAMP**ling **COD**e: 0: Sampling for positive transition of DI. 1: Sampling for negative transition of DI 2: Sampling when coincidence occurs. 3: Sampling when order is given from the simultaneous sampling function (C1).
40	ACTSS	IB	**ACT**ivates **S**imultaneous **S**ampling of the channels defined by terminals 41 to 44. Selected by call parameter C1.
41	SELSP1	IB	**SEL**ect simultaneous sampling of **SP**eed value channel **1**. Selected by call parameter C1. Only valid if terminal ACTSS above also is activated.
42	SELPULS1	IB	**SEL**ect simultaneous sampling of **PULS**e counter channel **1**. Selected by call parameter C1. Only valid if terminal ACTSS above also is activated.
43	SELSP2	IB	**SEL**ect simultaneous sampling of **SP**eed value channel **2**. Selected by call parameter C1. Only valid if terminal ACTSS above also is activated.

Number	Name	Type	Description
44	SELPULS2	IB	**SEL**ect simultaneous sampling of **PULS**e counter channel **2**. Selected by call parameter C1. Only valid if terminal ACTSS above also is activated.

Function

Please refer to DP820-I

ERROR HANDLING

The following errors may appear on the ERRTYPE output terminal:

0 OKAY, no error.
1 Data Base element not found
2 Data base element not implemented
3 Data base element not in service
4 Illegal Data base element TYPE property
5 Data base error
7 Access error (including bus error)
8 Parameter error
9 Invalid channel number
10 Illegal channel for Simultaneous Sampling

If C1 = 1 while input terminal CHANNEL is set to 2 no Error is given but only the Error code: "Illegal channel for simultaneous sampling".

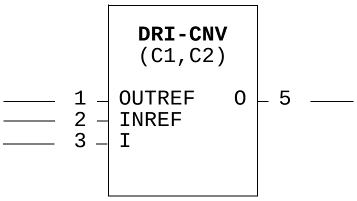
If the Coincidence Set Point is outside its range it is limited, no Error is given but only the Error code: "Parameter error".

Convert Data with Scale Factor DRI-CNV

Summary

DRI-CNV (DRIve CoNVert) converts data to from integer and real by using a scaling algorithm.

DRI-CNV is useful for conversion of data received by DRI-R or sent by DRI-S.



Call DRI-CNV(C1, C2,)

Parameter	Significance	Permissible values
C1	Input data type	I, R
C2	Output data type	I, R

Terminal Description

No	Name	Type	Description
1	OUTREF	I*C1*	Value of **OUT REF**erence which corresponds to value defined at in reference. This value is used for scaling, see formula below.
2	INREF	I*C2*	Value of **IN REF**erence which corresponds to the value defined at out reference. This value is used for scaling, see formula below.
3	I	I*C1*	**I**nput, Input data to be converted
5	O	O*C2*	**O**utput, Output for converted data

Function

Data input I with data type accordance with call parameter *C1* is converted to data with data type accordance to call parameter *C2* at the output O. The conversion is using the following formula for scaling.

$$O = \frac{I \times OUTREF}{INREF}$$

When conversion is to the data type I, overflow can occur at the max or min limit values. The output signal is then limited to the upper and lower limits of integer data (-32768 - 32767).

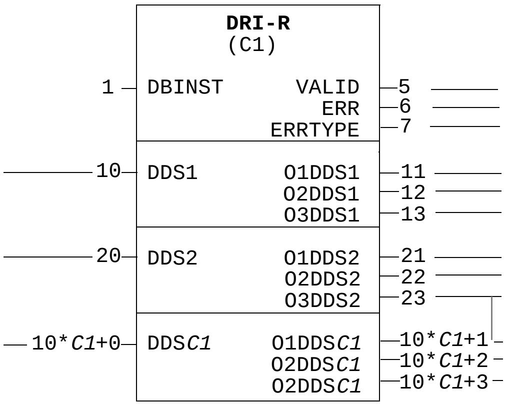
The quotient of the out reference and in reference is calculated only at every 20:th. cycle to reduce the load on the element. The result is buffered and used in the rest of the calculation. The OUTREF and INREF should therefore not be dynamic.

Receive Drive Data on Advant Fieldbus 100 **DRI-R**

Summary

DRI-R (DRIve - Receive) is used for receiving drive cyclic data on Advant Fieldbus 100. Depending on the drive, different data base elements are used for configuration of the communication characteristics. The data base element can be one of the following types.

DRISTD - standard drive connected to a S800 I/O station.
DRIENG - engineered drive connected to a S800 I/O station.



Call DRI-R(C1)

Parameter	Significance	Permissible values
C1	Number of Drive Data Sets	1 - 10

Terminal Description

No	Name	Type	Description
1	DBINST	II	**D**ata **B**ase **INST**ance. The instance number of DRISTD or DRIENG in the S800 interface file, alternatively the name of the DRISTD or DRIENG instance.
5	VALID	OB	**Valid** data is present
6	ERR	OB	**ERR**or flag
7	ERRTYPE	OI	**ERR**or **TYPE**
10	DDS1	II	**D**rive **D**ata **S**et number for **1**:st group of data
11	O1DDS1	OI	**O**utput **1** from **D**rive **D**ata **S**et **1**
12	O2DDS1	OI	**O**utput **2** from **D**rive **D**ata **S**et **1**
13	O3DDS1	OI	**O**utput **3** from **D**rive **D**ata **S**et **1**
...			
10**C1* +0	DDS*C1*	II	**D**rive **D**ata **S**et number for *C1* group of data
10**C1* +1	O1DDS*C1*	OI	**O**utput **1** from **D**rive **D**ata **S**et *C1*
10**C1* +2	O2DDS*C1*	OI	**O**utput **2**from **D**rive **D**ata **S**et *C1*
10**C1* +3	O3DDS*C1*	OI	**O**utput **3**from **D**rive **D**ata **S**et *C1*

Function

DRI-R is used for receiving cyclic drive data over Advant Fieldbus. Data of type Integer is supported. The drive is connected to an S800 I/O station. The DBINST terminal refers to the data base element DRISTD or DRIENG. The data to the drive are only in integer format. Conversion PC-element DRI-CNV converts integer data to real with proper scaling.

You specify the number of Drive Data Set to receive data from with call parameter C1. The DDSx terminal specifies from which Drive Data Set the group of 3 output values are received from.

When drives are connected to controllers (AC 70 or APC2), PC-element DSP-R is used together with DSP data base elements.

Configuration

The configuration work involved depends on how the drive is connected:

1. Drive connected to S800 I/O station

The communication is configured by means of DRISTD or DRIENG data base element. The DRISTD is used for standard drive hardware and DRIENG is used for engineered drives. The data base elements are configured according to the DB element specification.

The terminal IOSCANT configures the cycle time for data to be transfer on the bus for standard drives (in ms).
Allowed values are specified in the DB element specification.

The address of the drive is specified by BUS, STATION and POSITION.

2. Drive connected to controller (AC70 or APC2)

See the description of PC-element DSP-R.

Valid

The output terminal VALID indicates if the values available on the data output terminals are updated. The VALID terminal indicates as soon as one DDS has invalid data.

Error handling

The following error codes can be reported on the ERRTYPE terminal:

0 OK
1 Data base element according to DBINST is not found. The data base element could be DRISTD or DRIENG.
3 Error detected by corresponding data base element or illegal DB configuration
4 Bus Error. The AF 100 interface module is not reachable
5 Data set number out of range

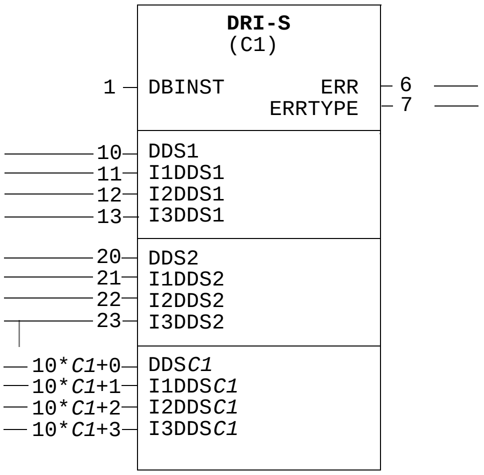
The terminal ERR is set to 1 if ERRTYPE <> 0.

Send Drive Data on Advant Fieldbus 100 DRI-S

Summary

DRI-S (DRIve - Send) is used for sending drive cyclic data on Advant Fieldbus 100. Depending on the drive different data base elements are used for configuration of the communication characteristics. The data base element can be one of the following types.

DRISTD - standard drive connected to a S800 I/O station.
DRIENG - engineered drive connected to a S800 I/O station.



Call DRI-S(C1)

Parameter	Significance	Permissible values
C1	Number of Drive Data Sets	1 - 10

Terminal Description

No	Name	Type	Description
1	DBINST	II	**D**ata **B**ase **INST**ance. The instance number of DRISTD or DRIENG in the S800 interface file, alternatively the name of the DRISTD or DRIENG instance.
6	ERR	OB	**ERR**or flag
7	ERRTYPE	OI	**ERR**or **TYPE**
10	DDS1	II	**D**rive **D**ata **S**et number for **1**:st group of data
11	I1DDS1	II	**I**nput **1** to **D**rive **D**ata **S**et **1**
12	I2DDS1	II	**I**nput **2** to **D**rive **D**ata **S**et **1**
13	I3DDS1	II	**I**nput **3** to **D**rive **D**ata **S**et **1**
...			
10**C1* +0	DDS*C1*	II	**D**rive **D**ata **S**et number for *C1* group of data
10**C1* +1	I1DDS*C1*	II	**I**nput **1** to **D**rive **D**ata **S**et *C1*
10**C1* +2	I2DDS*C1*	II	**I**nput **2**to **D**rive **D**ata **S**et *C1*
10**C1* +3	I3DDS*C1*	II	**I**nput **3**to **D**rive **D**ata **S**et *C1*

Function

DRI-S is used for sending cyclic drive data over Advant fieldbus. Data of type Integer is supported. The drive is connected to a S800 I/O station. The DBINST terminal refers to the data base element DRISTD or DRIENG. The data to the drive are only in integer format. Conversion PC-element DRI-CNV converts real data to integer which are transmitted on the bus.

The user specifies the number of Drive Data Set to send data to with call parameter C1. The DDSx terminal specifies to which Drive Data Set the group of 3 input values are sent to.

When drives are connected to controllers (AC 70 or APC2) PC-element DSP-S are used together with DSP data base elements.

Configuration

The configuration work involved, depending how the drive is connected are as follow:

1. **Drive connected to S800 I/O station**

The communication is configured by means of DRISTD or DRIENG data base element. The DRISTD is used for standard drive hardware and DRIENG is used for engineered drives. The data base elements are configured according to the DB element specification.

The terminal IOSCANT configures the cycle time for data to be transfer on the bus for standard drives (in ms). Allowed values are specified in the DB element specification.

The address of the drive is specified by BUS, STATION and POSITION.

2. **Drive connected to controller (AC70 or APC2)**

See the description of PC-element DSP-S.

Error handling

The following error codes can be reported on the ERRTYPE terminal:

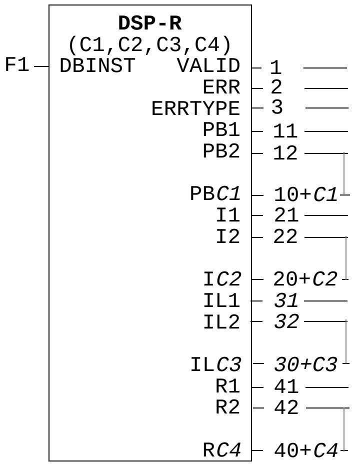
0 OK
1 Data base element according to DBINST is not found. The data base element could be DRISTD or DRIENG.
3 Error detected by corresponding data base element or illegal DB configuration
4 Bus Error. The AF100 interface module is not reachable
5 Data set number out of range.

The terminal ERR is set to 1 if ERRTYPE <> 0.

Receive Cyclic Data on Advant Fieldbus 100 DSP-R

Summary

DSP-R (Data Set Peripheral -Receive) is used to receive cyclic data on Advant Fieldbus 100. It uses the Data Set Peripheral data base element for configuration of the communication characteristics.



Call DSP-R(C1, C2, C3, C4)

Parameter	Significance	Permissible values(1)
C1	Number of Boolean block words	0 - 8
C2	Number of Integers	0 - 8
C3	Number of Long Integers	0 - 8
C4	Number of Real numbers	0 - 8

(1) Call Parameter Relation:
$0 < (C1 + C2 + C3 + C4) < 9$

Terminal Description

No	Name	Type	Description
F1	DBINST	FI	Reference to corresponding DSP **D**ata **B**ase element. **INST**ance number.
10	VALID	OB	**Valid** data is present
11	ERR	OB	**ERR**or flag
12	ERRTYPE	OI	**ERR**or **TYPE**
11	PB1	OIL	**P**acked **B**oolean block word **1** (32 bits)
... 10+*C1*	PB*C1*	OIL	
21	I1	OI	**I**nteger data **1**
... 20+*C2*	I*C2*	OI	

Terminal Description

No	Name	Type	Description
31	IL1	OIL	**I**nteger **L**ong data **1**
... 30+*C3*	IL*C3*	OIL	
41	R1	OR	**R**eal data **1**
... 40+*C4*	R*C4*	OR	

Function

DSP-R is used for reception of cyclic data over Advant Fieldbus 100. Data of type Boolean block words, Integer, Integer Long and Real are supported. If Boolean is selected, PC element CONV-IB(IL,4,31) is recommended for unpacking from type IL.

Data received by DSP-R could be sent from other stations either with PC element DSP-S or with data base element DSP (USER = 0 and with the related DAT elements).

Configuration

The communication is configured by means of Data Set Peripheral (DSP) data base element.

The DSP element should be created with USER = 4, SOURCE = RECEIVE and NO_BREC, NO_INT, NO_INTL, NO_REAL set to zero. This excludes the ordinary scanning performed from DSP.

The identity of the data set is defined by terminals IDENT, BUS and STATION.

It is important that the configuration by means of data types is the same in both ends. If data base element DSP is used in other node for sending of data, it must be created with SORT_REF = YES, and the sum of NO_BREC, NO_INT, NO_INTL and NO_REAL must be at least 5.

Valid

The output terminal VALID indicates if the values available on the data output terminals are updated. For proper function of VALID the CYCLETIM parameter in corresponding data base element must be set to the same time (in ms) as defined by the sender. If CYCLETIM is not equal, the time-out supervision indicated by terminal VALID will give false information. Allowed values for CYCLETIM are 1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024, 2048 and 4096.

Error handling

The following error codes can be reported on the ERRTYPE terminal:

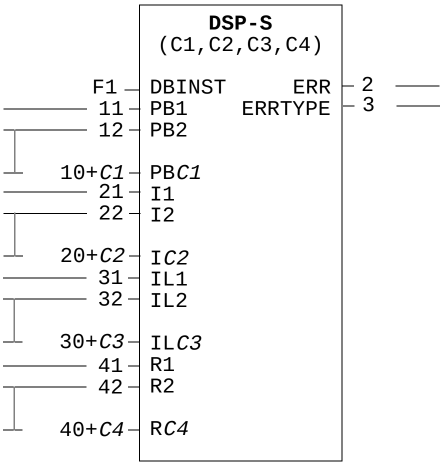
0 OK
1 Data base element DSP according to DBINST is not found
2 Corresponding DSP data base element has wrong USER value
3 Error detected by corresponding DSP data base element or illegal DB configuration
4 Bus Error. The AF 100 interface module is not reachable

The terminal ERR is set to 1 if ERRTYPE is <> 0.

Send Cyclic Data on Advant Fieldbus 100 DSP-S

Summary

DSP-S (Data Set Peripheral - Send) is used for sending cyclic data on Advant Fieldbus 100. It uses the Data Set Peripheral data base element for configuration of the communication characteristics.



Call DSP-S(C1, C2, C3, C4)

Parameter	Significance	Permissible values(1)
C1	Number of Boolean block words	0 - 8
C2	Number of Integers	0 - 8
C3	Number of Long integers	0 - 8
C4	Number of Real numbers	0 - 8

(1) Call Parameter Relation:
$0 < (C1 + C2 + C3 + C4) < 9$

Terminal Description

No	Name	Type	Description
F1	DBINST	FI	Reference to corresponding DSP **D**ata **B**ase element. **INST**ance number.
2	ERR	OB	**ERR**or flag
3	ERRTYPE	OI	**ERR**or **TYPE**
11	PB1	IIL	**P**acked **B**oolean block word **1** (32 bits)
... 10+*C1*	PB*C1*	IIL	
21	I1	II	**I**nteger data **1**
... 20+*C2*	I*C2*	II	
31	IL1	IIL	**I**nteger **L**ong data **1**

Terminal Description

No	Name	Type	Description
... 30+*C3*	IL*C3*	IIL	
41	R1	IR	**R**eal data **1**
... 40+*C4*	R*C4*	IR	

Function

DSP-S is used for sending cyclic data over Advant Fieldbus 100. Data of type Boolean block words, Integer, Integer Long and Real are supported. If Boolean is selected, PC element CONV-BI(IL,5,31) is recommended for packing to type IL (boolean block word).
Data sent from DSP-S could be received in other stations either with PC element DSP-R or with data base element DSP (USER = 0 and with the related DAT elements).

Configuration

The communication is configured by means of Data Set Peripheral (DSP) data base element.

The DSP element should be created with USER = 4, SOURCE = SEND and with NO_BREC, NO_INT, NO_INTL, NO_REAL set to zero. This inhibits the ordinary scanning performed from DSP.

The terminal CYCLETIM controls the cycle time for data to be transferred on the bus (in ms). Allowed values are 1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024, 2048 and 4096.

The identity of the data set is defined by terminals IDENT, BUS and STATION.

It is important that the configuration by means of data types is the same in both ends. If you use data base element DSP in other node for reception of data, you must have created it with SORT_REF = YES. And the sum of NO_BREC, NO_INT, NO_INTL and NO_REAL must be at least five.

Error handling

The following error codes can be reported on the ERRTYPE terminal:

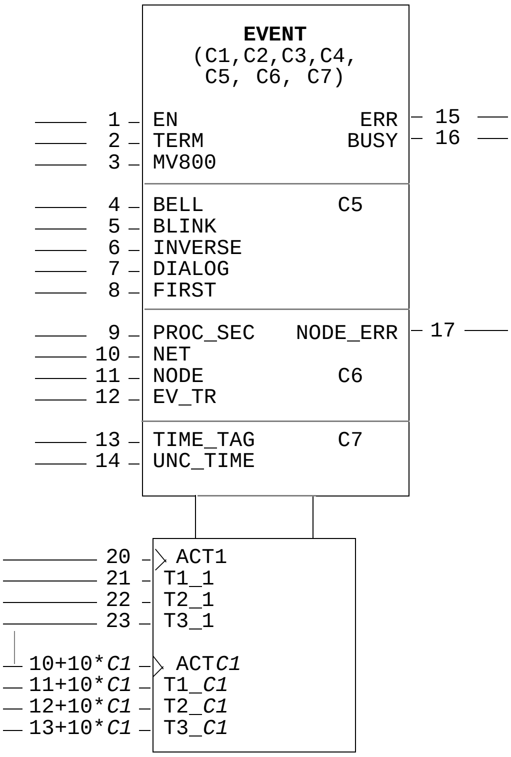
0 OK
1 Data base element DSP according to DBINST is not found
2 Corresponding DSP data base element has wrong USER value
3 Error detected by corresponding DSP data base element or illegal DB configuration
4 Bus Error. The AF 100 interface module is not reachable.

The terminal ERR is set to 1 if ERRTYPE <> 0.

Event Sending EVENT

Summary

EVENT is used to send events to the local operator station MasterView 320. An event consists of up to three subtexts along with the date and time. Though the function is limited, the element may also be used to send events to a central operator station (Advant Station 500 Series, MasterView 800 Series or SuperView).



PC element EVENT

Call EVENT (C1, C2, C3, C4, C5, C6, C7)

Parameter	Significance	Permissible values
C1	Number of events	1 - 9
C2	Number of characters on input T1	0 - 72
C3	Number of characters on input T2	0 - 72
C4	Number of characters on input T3	0 - 72
C5	Selection of extra connections for MV 320 0: Connections 4 - 8 not selected 1: Connections 4 - 8 selected	0 - 1
C6	Selection of extra connections for operator station. 0: Connections 9-12 and connection 17 not selected 1: Connections 9-12 and connection 17 selected	0 - 1
C7	Selection of connections for time marking 0: Connections 13 and 14 not selected 1:Connections 13 and 14 selected.	0 - 1
$C2+C3+C4>0$		

Terminal Description

No	Name	Type	Description
1	EN	IB	**EN**able. Input for activation of the element.
2	TERM	IA17	**TERM**inal. Input for selection of the MV 320 terminals to which the event is to be sent.
3	MV800	IB	**M**aster**V**iew **800**. Indicates that the event is to be sent to a central operator station.
4	BELL	IB	Input to get an audible signal when sending to an MV 320 terminal.
5	BLINK	IB	Input to get blinking text on an MV 320 terminal.
6	INVERSE	IB	Input to get inverted video on an MV 320 terminal.
7	DIALOG	IB	Input used to get an event that is not saved in the event list and that does not contain date and time on an MV 320 terminal.
8	FIRST	IB	If FIRST is set to 1, only the first active text group is sent to the selected terminals.
9	PROC_SEC	II	Indicates the process section in the operator station that the event belongs to.
10	NET	II	Network number of the node that the event is to be sent to.
11	NODE	II	Node number of the node that the event is to be sent to.
12	EV_TR	II	Input specifying pointer to a record in the file EVENT_TREAT.
13	TIME_TAG	IIL	Input for time-tagging.
14	UNC_TIME	IB	**UNC**ertain **TIME**. Used to tag uncertain time marking.
15	ERR	OB	**ERR**or. Output which, when it is set to 1, indicates a transmission error.
16	BUSY	OB	Output which, when set to 1, indicates that the receiver is busy.
17	NODE_ERR	OB	**NODE_ERR**or. Output which, when set to 1, indicates an error in communication with other nodes.
20	ACT1	IB	**ACT**ivate 1. Input to send event 1. Transmission takes place when the input becomes set to 1 after having been 0.
21	T1_1	IA*C2*	**T**ext **1** event **1**. Input for the first subtext of event 1. The text starts in position 1 on the event line.
22	T2_1	IA*C3*	**T**ext **2** event **1**. Input for the second subtext of event 1. The text starts in position 21 on the event line.
23	T3_1	IA*C4*	**T**ext **3** event **1**. Input for the third subtext of event 1. The text starts in position 46 on the event line.
: 10+10x*C1*	ACT*C1*	IB	**ACT**ivate ***C1***. Input to send event *C1*. Transmission takes place when the input becomes set to 1 after having been p.
11+10x*C1*	T1_*C1*	IA*C2*	**T**ext **1** event ***C1***. Input for the first subtext of event *C1*. The text starts in position 1 on the event line.
12+10x*C1*	T2_*C1*	IA*C3*	**T**ext **2** event ***C1***. Input for the second subtext of event *C1*. Input for the second subtext of event *C1*. The text starts in position 21 on the event line.
13+10x*C1*	T3_*C1*	IA*C4*	**T**ext **3** event ***C1***. Input for the third subtext of event *C1*. The text starts in position 46 on the event line.

Function

For an event to be sent, the input EN must be set to 1. Transmission then takes place when one or more ACT inputs becomes 1, after having been 0. Each text group is activated independently of the others. If any terminal cannot receive the events sent to it, this is indicated by the output BUSY being set to 1. The element then sends the same event at each execution until the transmission succeeds, at which time the output BUSY is reset to 0. While BUSY is set to 1 new events are lost. The inputs ACT, however, are decoded at every execution.

Date and time of the event is system time when the element is executed. If connections for time marking are selected time is read from connection 13, TIME_TAG.

MasterView 320

An event is shown on MV 320 in a line of 80 characters. The three subtexts T1, T2 and T3 start in position 1, 21 and 46 respectively. If any of the subtexts are longer than the text field in which it is to be written (for example, if T1 is longer than 20 characters), the following field will overwrite the excess characters. The last 22 positions on the line are occupied by date and time. Example 1 The appearance of the text line on MV 320:

D=’Text 1’ ------21 │T1_1

D=’Text 2’ ------22 │T2_1

D=’Text 3’ ------23 │T3_1

gives the following line on MV 320:

Text 1Text 2Text 31990-02-18/16:10:15.2

A text string on the input TERM indicates which MV 320 terminals are to receive the events. All digits appearing in the string are translated to the corresponding terminal, while all other characters (alphabetical characters, separators) are ignored.

Example 2 Sending to MV 320 terminal 1

D=’Terminal 1’ ------2 │TERM

Extra Connections for MasterView 320

Select extra connections for MV 320 by setting the call parameter *C5* to 1.

BELL gives a tone signal when the event line first is printed.

BLINK gives an intermittent shown event line.

INVERSE gives an event line with inverted video.

DIALOG is used in operator dialogs. Date and time are not sent, and the vent is not saved in the event list. You may use DIALOG to erase the event line by sending a blank line. If sending is to the operator station, DIALOG must be 0.

FIRST is used to prioritize events. If FIRST is set to 1, at most one event is sent during each execution. The element scans from event 1 and forwards and sends only the first active event found. You may also use this function when sending to the operator station.

Central Operator Station

An event to be sent to an operator station (Advant Station 500 Series, MasterView 800 or SuperView) consists of two subtexts along with date and time. The subtexts consist of the first twenty characters from inputs T1 and T2. Events are sent to the operator station terminals if the input MV 800 is set to 1.

A remote computer is addressed in the same way as SuperView.

If you do not select extra connections for the operator station, that is, if *C6* is set to 0, the events are sent to all the operator stations in the own control network.

Extra Connections for the Central Operator Station

Select extra connections for the operator station by setting call parameter *C6* to 1. These connections do not affect MV 320. PROC_SEC indicates the process section in the operator station that the event belongs to. Permitted values are -1 - 16. If other values are set, the process section is set to 0, that is, the event is displayed on all terminals at the receiving node. If the value -1 is entered, the event is not displayed on any terminal.

NET and NODE indicate the receiving node. If NET or NODE is set to 0, the events are sent to all the operator station nodes in the own control network.

EV_TR indicates a pointer to a record in the file EVENT_TREAT. Value 14 corresponds to the normal function of the EVENT element.

NODE_ ERR

On sending to own control network: NODE_ERR is set to 1 if the sending node has no contact with the node specified by NET and NODE.

On sending to another network: NODE_ERR is set to 1 if the sending node has no contact with the network indicated by NET. If NET or NODE is set to 0, that is, if no receiving node is specified, NODE_ERR is set to 0.

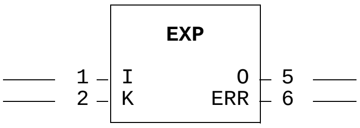
Connections for Time Marking

Select connections for time marking by setting call parameter C7 to 1. These connections are used to time-tag events received from a remote computer, for example via RCOM. TIME_TAG gives the time in 0.1 ms from midnight. The data type is IntegerLong. Data of types T or TR cannot be connected to this input. TIME_TAG can be connected to status word status 3 of RCOM. UNC_TIME is used to mark uncertain time tagging. When UNC_TIME is set to 1, the letter ’U’ appears after the time in the event text on the operator station. On MV 320, a ’u’ appears before the time. UNC_TIME can be connected to RCOM status1 bit 9.

Exponentiation EXP

Summary

EXP (**EXP**onential) is used to calculate the exponential of a real number. The result can be multiplied by an optional value.



PC element EXP

Call EXP

Terminal Description

No	Name	Type	Description
1	I	IR	**I**nput. Input for number of which the exponential is to be calculated.
2	K	IR	Input for multiplication factor K.
5	O	OR	**O**utput. Output for the product of the value at input K and the exponential of the value at input I.
6	ERR	OB	**ERR**or. Output which is set to 1 if the value at output O is greater than the largest or less than the smallest value that can be represented.

Function

The exponential of the value at input I is calculated. The result is multiplied by the value at the input K. The product is stored at the output O.

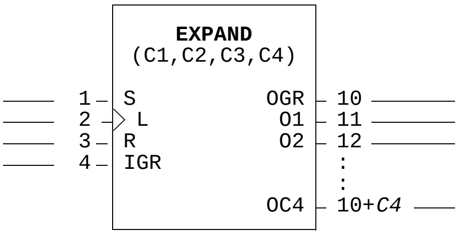
Supervision

If the value to be stored at output O is greater than the largest, or less than the smallest, value that can be represented, the error output ERR is set (to 1).

Group Data Expander EXPAND

Summary

The PC element EXPAND expands the values in the group data to normal data types.



PC element EXPAND

Call EXPAND (C1, C2, C3, C4)

Parameter	Significance	Permissible values
C1	Data type	I, IL, R, B, T, TR
C2	Number of positions at the input in group data	1 - 255
C3	Number of positions at the output in group data	0 - 255
C4	Number of items of data to be expanded to outputs 3+*C4*=< *C2*	1 - 30

Terminal Description

No	Name	Type	Description
1	S	IB	**S**et. Input for expanding of values, each time the element is executed.
2	L	IB	**L**oad. Dynamic input for expanding of values.
3	R	IB	**R**eset. Resetting of the outputs.
4	IGR	IG*C2C1*	**I**n **GR**oup. Input data of group type.
10	OGR	OG*C3C1*	**O**ut **GR**oup. Output data of group type.
11	O1	O*C1*	**O**ut **1**. First expanded output data.
12	O2	O*C1*	**O**ut **2**. Second expanded output data.
: 10+ *C4*	O*C4*	O*C1*	**O**ut ***C4***. *C4*th expanded output data.

Function

EXPAND reads an input of the type ’Group data’ and copies the number of elements specified by call parameter *C4* to the expansion outputs and continues to copy subsequent data to the group output. The first part of the group will be expanded first. Having no elements at the group output is one of the options. Parameter *C3* determines where the expansion ends. To expand group data with more than 30 elements, several EXPAND elements are connected in series. The element expands data to the outputs when input L goes to 1 or on each execution if input S is set.

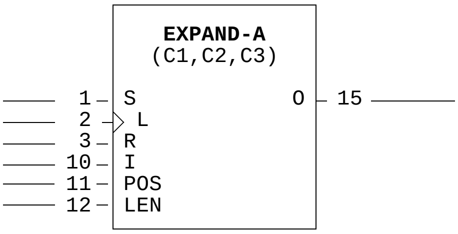
Resetting

All outputs are reset when input R is set.

Byte Array Data Expander EXPAND-A

Summary

EXPAND-A (**EXPAND A**rray) extracts data from a Byte Array and puts the extracted data on the output. The output can be of data types Array, 16 bit integer, 32 bit integer, or Real Number.



PC element EXPAND-A

Call EXPAND-A (C1, C2, C3)

Parameter	Significance	Permissible values
C1	Data type for the output	I, IL, R, A
C2	Number of characters at the input array.	1 - 80
C3	Number of characters at the output if *C1*=A	0 - 80

Terminal Description

No	Name	Type	Description
1	S	IB	**S**et. Input for expanding of values, each time the element is executed.
2	L	IB	**L**oad. Dynamic input for expanding of values.
3	R	IB	**R**eset. Input for resetting of the outputs.
10	I	IA*C2*	**I**n. Input for byte array.
11	POS	II	**POS**ition. Specifies in which position the extraction is to begin (number of characters omitted +1).
12	LEN	II	**LEN**gth. Specifies how many characters should be extracted from the array.
15	O	O*C1*	**O**ut. Output for extracted data.

Function

EXPAND-A extracts a number of characters, specified by input terminals POS and LEN, from the input array. The extracted data is stored at the output terminal O. If the output is of data type array, the whole array first is filled by ASCII code for ”space” and after that, LEN number of characters are copied from the input array.

If the output data type is I, IL or R, LEN number of characters are taken from the input array, transformed from an ASCII representation to a numerical representation and stored at the output. Allowed non digit signs in the beginning of input array are ”space”, ”plus”, and ”minus” signs. ”Plus” and ”minus” signs must be followed by a digit. Number of ”space” are not limited. Input S overrides L, so when input S is set, L has no effect.

Normal Execution

When input S is set, data is continuously moved to the output terminal. The element extracts data to the output terminals when input terminal L goes from 0 to 1.

Clearing

When input R is set, all data at the outputs are cleared. Data type A is set to the ASCII representation for ”space”. Data type I and IL is set to zero. R overrides both S and L.

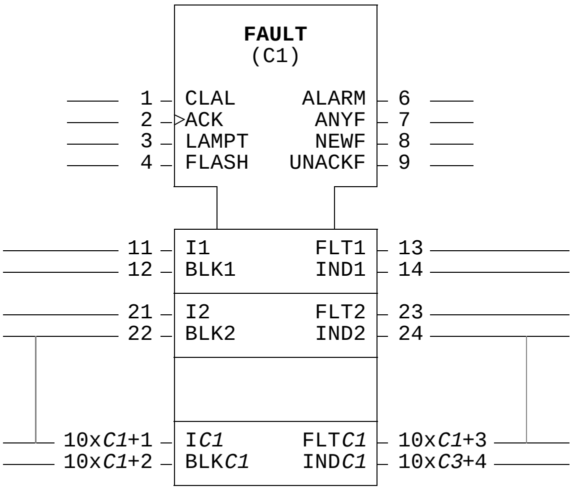
Fault Handling

If input POS is negative or greater than parameter *C2* the output is cleared. If input POS plus input LEN are greater than *C2*, *C2*, minus value of POS, is used in the conversion instead of input LEN. If the output is of data type I or IL, and an illegimate sign is found in the part which shall be extracted, the output is set to zero.

Error Signal Handling FAULT

Summary

The error signal element FAULT indicates errors by means of flashing light signals. You can use it for maximum nine error signals. Common outputs are provided for indication of errors in the signal groups.



PC element FAULT

Call FAULT (C1)

Parameter	Significance	Permissible values
C1	Number of error signal points	1 - 9

Terminal Description

No	Name	Type	Description
1	CLAL	IB	**CL**ear **AL**arm. Input for canceling alarm.
2	ACK	IB	**ACK**nowledge. Input for acknowledgement of error signal.
3	LAMPT	IB	**LAMP T**est. Input for lamp test.
4	FLASH	IB	Input for flash signal.
6	ALARM	OB	**ALARM** output.
7	ANYF	OB	**ANY F**ault. Common output for indication of remaining error.
8	NEWF	OB	**NEW F**ault. Output which gives a pulse during one program cycle each time a new error occurs.
9	UNACKF	OB	**UNACK**nowledged **F**ault. Output set to 1 while any error within the element remains unacknowledged.
11	I1	IB	**I**nput **1**. Error signal input for the first signal point.
12	BLK1	IB	**BL**oc**K 1**. Blocking input for the first signal point.
13	FLT1	OB	**F**au**LT 1**. Output set to 1 while the error signal for the first signal point remains.

Terminal Description

No	Name	Type	Description
14	IND1	OB	**IND**ication **1**. Error signal output to the signal lamp for the first signal point.
21	I2	IB	**I**nput **2**. Error signal input for the second signal point.
22	BLK2	IB	**BL**oc**K 2**. Blocking input for the second signal point.
23	FLT2	OB	**F**au**LT 2**. Output which is set to 1 while the error signal for the second signal point remains.
24	IND2	OB	**IND**ication **2**. Error signal output to the signal lamp for the second signal point.
: 10x*C1*+1	I*C1*	IB	**I**nput ***C1***. Error signal input for the last signal point in the element.
10x*C1*+2	BLK*C1*	IB	**BL**oc**K *C1***. Blocking input for the last signal point in the element.
10x*C1*+3	FLT*C1*	OB	**F**au**LT *C1*.** Output which is set to 1 while the error signal for the last signal point in the element remains.
10x*C1*+4	IND*C1*	OB	**IND**ication ***C1***. Error signal output to the signal lamp for the last signal point in the element.

Function

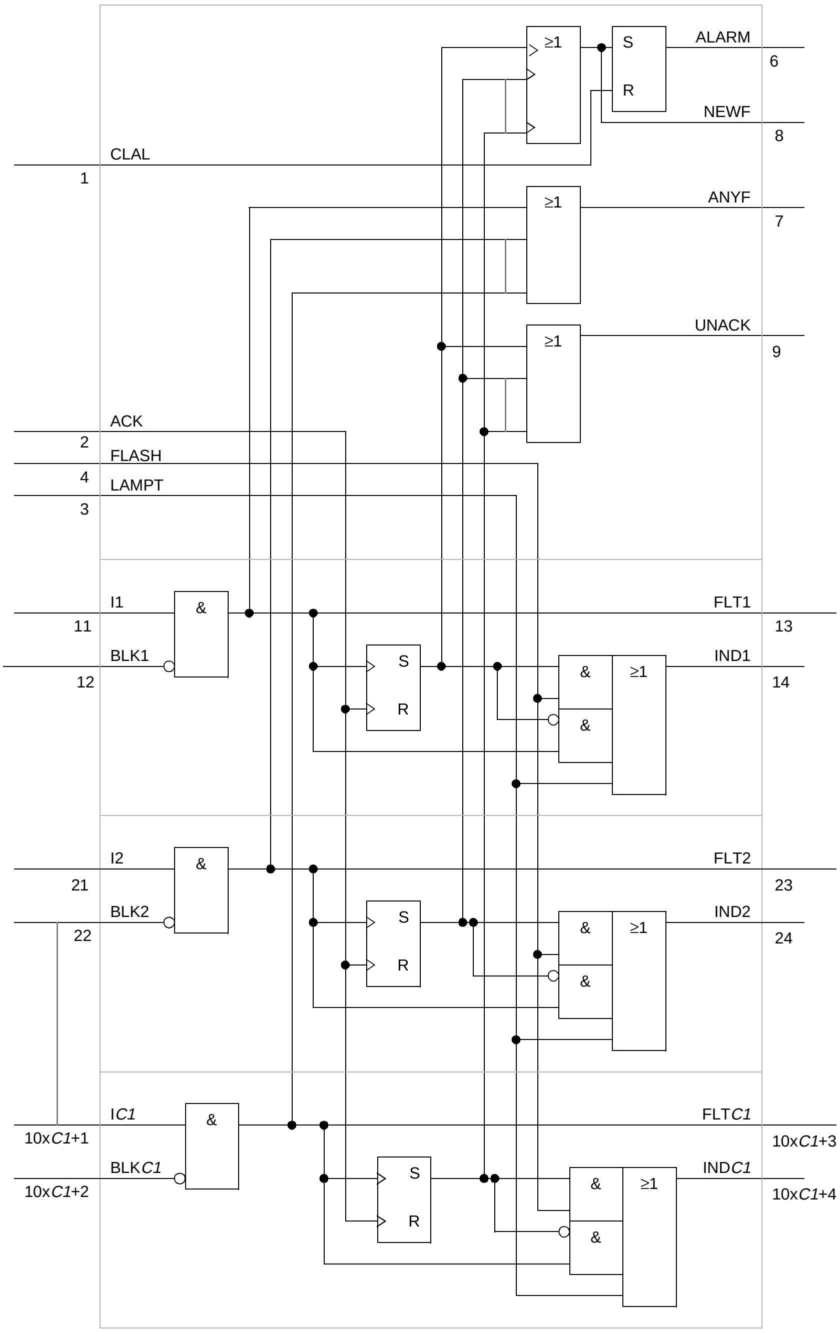
When an error occurs, an alarm is activated at the same time as the error indication. An alarm is obtained irrespective of whether previous errors indicated by other signal points have been acknowledged or not. The alarm is canceled via the input CLAL.
If a signal point is not acknowledged, and the error status at this signal point first ceases and then reappears, no new alarm is obtained. Before the acknowledgment, the light flashes with the frequency of the FLASH input. After acknowledgment, the indication remains steady until the error ceases.

Blocking of Error Signals

The signals at the inputs I1 - I*C1* may be locked by connecting a blocking signal to the inputs BLK1 - BLK*C1*.

Common Signals

Common signals are provided for remaining errors (ANYF) and unacknowledged errors (UNACKF) at any of the error signal points. A pulse with the duration of one program cycle is obtained at the NEWF output when a new error occurs.

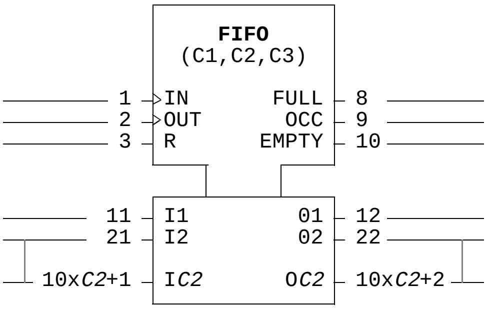


Function diagram

Queue Register FIFO

Summary

The queue register FIFO (**F**irst **I**n **F**irst **O**ut) stores data to use later, irrespective of when the it was stored. The queues work in parallel, that is, reading in, reading out, etc. are performed at the same time in all queues. Data may be integer, real number, Boolean or time. You can select up to nine queues with maximum 64 queue places each.



PC element FIFO

Call FIFO (C1, C2, C3)

Parameter	Significance	Permissible values
C1	Data type	I, IL, R, B, T, TR
C2	Number of queues	1 - 9
C3	Number of queue places in each queue	2 - 64

Terminal Description

No	Name	Type	Description
1	IN	IB	Dynamic input for entry of data in the queues. Data is read from the inputs I1 - I*C2*.
2	OUT	IB	Dynamic input for reading data from the queues.
3	R	IB	**R**eset. Input which clears the queue register and prevents all further entry.
8	FULL	OB	Output which is set when the queue register is full.
9	OCC	OI	**OCC**upied. Output which specifies how many queue places are occupied.
10	EMPTY	OB	Output which is set when the queue register is empty.
11	I1	I*C1*	**I**nput **1**. Input data to queue no. 1.
12	O1	O*C1*	**O**utput **1**. Output data from queue no. 1.
21	I2	I*C1*	**I**nput **2**. Input data to queue no. 2.
22	O2	O*C1*	**O**utput **2**. Output data from queue no. 2.
: 10x*C2*+1	I*C2*	I*C1*	**I**nput ***C2***. Input data to queue no. *C2*.
10x*C2*+2	O*C2*	O*C1*	**O**utput ***C2***. Output data from queue no. *C2*.

Function

Each queue has a data input I, which is used when data is to be entered at the end of the queue. Each queue has a data output O. Data from place 1 in each queue is continuously available at outputs O1 - O*C2*. If both the dynamic inputs (IN and OUT) are set during the same program cycle, entry is performed first and then reading out.

Entry of Data

Data at the I inputs are placed last in the queues when input IN is set. When the queue register is empty, data is stored first at queue place 1, then at queue place 2, etc.

Output of Data

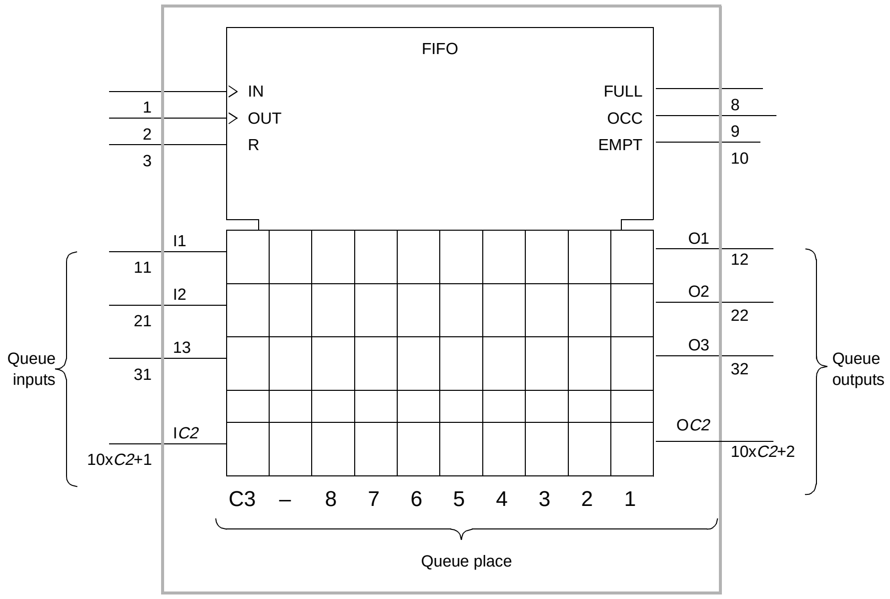
When input OUT goes to 1, data at each used place in all the queues is moved to the place with the number immediately below. 0 is written at the last place used. Data which was previously stored at queue place 2 is now at queue place 1, and can be read at outputs O1 - O*C2*.

Clearing of the Complete Register

When input R is set, the complete contents of the queues are cleared. All further entry is blocked.

Supervision

How many queue places are occupied is indicated continuously by output OCC. When the queue register is full, the output FULL is set. When the queue register is empty, the output EMPTY is set.

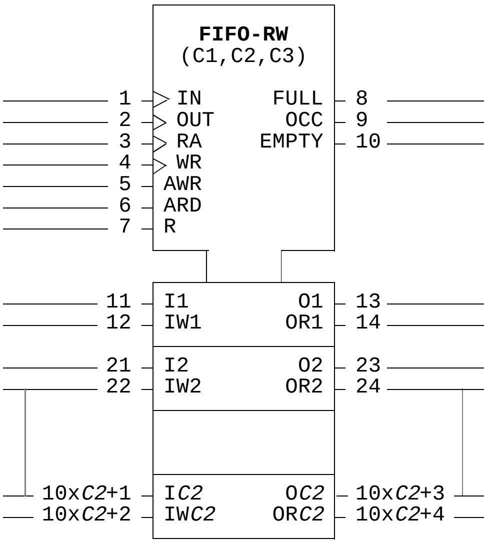


Function diagram

Queue Register, Read/ Write FIFO-RW

Summary

The queue register FIFO-RW (**F**irst **I**n **F**irst **O**ut **- R**ead **W**rite) stores data to use later, irrespective of when it was stored. Data in the queues can be readily changed and deleted at any queue place. New data is entered last in the queues. The queues in the element work in parallel, that is, reading in, reading out, etc. are performed at the same time in all queues. Data may be of integer, real number, Boolean or time. You can select maximum nine queues with maximum 64 queue places each.



PC element FIFO-RW

Call FIFO-RW (C1, C2, C3)

Parameter	Significance	Permissible values
C1	Data type	I, IL, R, B, T, TL
C2	Number of queues	1 - 9
C3	Number of queue places in each queue	2 - 64

Terminal Description

No	Name	Type	Description
1	IN	IB	Dynamic input for entry of data in the queues. Data is read from the inputs I1 - I*C2*.
2	OUT	IB	Dynamic input for reading data from the queues.
3	RA	IB	**R**eset **A**ddress. Dynamic input for clearing data at the queue place in all queues specified by the input AWR.
4	WR	IB	**WR**ite. Dynamic input for entry of data at the queue place in all queues specified by the input AWR. Data is accessed at the inputs IW1 - IW*C2*.
5	AWR	II	**A**ddress **WR**ite. Address of the queue place which is to be changed or cleared by WR or RA.
6	ARD	II	**A**ddress **R**ea**D**. Address of the queue place to be read.
7	R	IB	**R**eset. Input which clears the queue register and prevents all further entry.

Terminal Description

No	Name	Type	Description
8	FULL	OB	Output which is set when the queue register is full.
9	OCC	OI	**OCC**upied. Output which specifies how many queue places are occupied.
10	EMPTY	OB	Output which is set when the queue register is empty.
11	I1	I*C1*	**I**nput **1**. Input data to queue no. 1.
12	IW1	I*C1*	**I**nput **W**rite **1**. Input data to queue no. 1. Used when data is to be changed according to AWR, WR and RA.
13	O1	O*C1*	**O**utput **1**. Output data from queue no. 1.
14	OR1	O*C1*	**O**utput **R**ead **1**. Output data from queue no. 1. Used when data is to be read from queue places specified by the input ARD.
21	I2	I*C1*	**I**nput **2**. Input data to queue no. 2.
22	IW2	I*C1*	**I**nput **W**rite **2**. Input data to queue no. 2. Used when data is to be changed according to AWR, WR and RA.
23	O2	O*C1*	**O**utput **2**. Output data from queue no. 2.
24	OR2	O*C1*	**O**utput **R**ead **2**. Output data from queue no. 2. Used when data is to be read from queue places specified by the input ARD.
: 10x*C2*+1	I*C2*	I*C1*	**I**nput ***C2***. Input data to queue no. *C2*.
10x*C2*+2	IW*C2*	I*C1*	**I**nput **W**rite ***C2***. Input data to queue no. *C2*. Used when data is to be changed according to AWR, WR and RA.
10x*C2*+3	O*C2*	O*C1*	**O**utput ***C2***. Output data from queue no. *C2*.
10x*C2*+4	OR*C2*	O*C1*	**O**utput **R**ead ***C2***. Output data from queue no. *C2*. Used when data is to be read from queue places specified by the input ARD.

Function

Each queue has two data inputs, I and IW. I is used when data is to be entered last in the queues and IW when data is to be changed at an optional place in the queues. Each queue has two data outputs, O and OR. Data from place 1 in each queue is continuously available at outputs O1 - O*C2*. If two or more of the dynamic inputs (IN, OUT, RA and WR) are set during the same program cycle, data is first entered, then changed or deleted, and finally read out.

Entry of Data

Data at I inputs are entered last in the queues when input IN is 1. When the queue register is empty, data is stored first at queue place 1, then at queues place 2, and so on.

Readout and Reading of Data

When input OUT goes to 1, data at each used place in all of the queues is moved to the place with the number immediately below. 0 is written at the last place used. Data which was previously stored at queue place 2 is now at queue place 1 and can be read at the outputs O1 - O*C2*.

Changing Data at an Optional Queue Place

When WR goes to 1, the contents of all queues at the queue place pointed out by the input AWR is changed to the values at the data inputs IW1 - IW*C2*.

Reading of Data at an Optional Queue Place

Data from the queue place pointed out by the input ARD is continuously available at the data outputs OR1 - OR*C2*.

Deletion of Data in the Queues

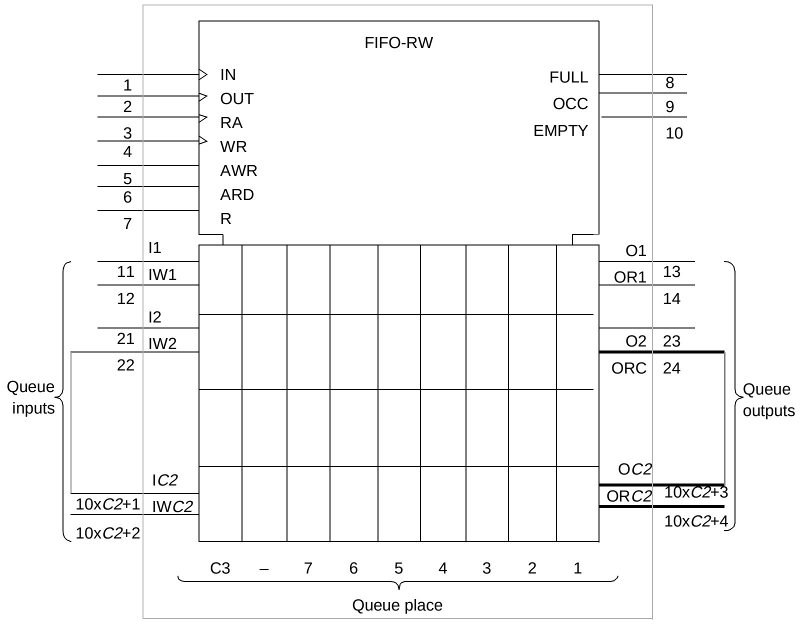
When input RA is set, data is removed form the place pointed out by the address input AWR. The contents at the places after that pointed out is moved forward one step and the last place previously used is cleared.

Clearing of the Complete Register

When the input R is set, the complete contents of the queues are cleared. All further entry is blocked.

Supervision

How many queue places are occupied is indicated continuously by output OCC. When the queue register is full, the output FULL is set and when the queue register is empty, the output EMPTY is set.

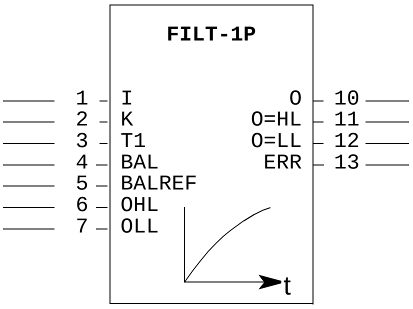


Function diagram

Filter with 1 pole FILT-1P

Summary

FILT-1P (**FILT**er **- 1 P**ole) is used as a single pole low pass filter. You can limit the output signal with limit values specified at special inputs. The balancing function permits the output signal to follow an external reference and permits a bumpless return to the normal function. All transfers from static states are bumpless.



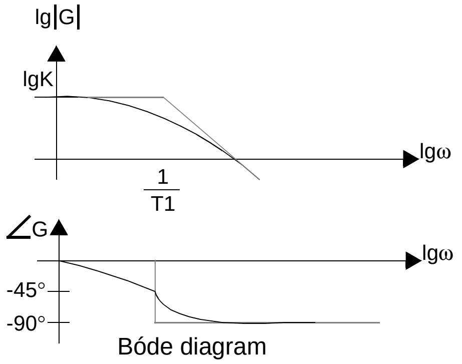
PC element FILT-1P

Call FILT-1P

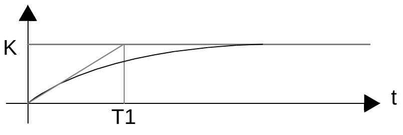
Terminal Description

No	Name	Type	Description
1	I	IR	**I**nput. Input for actual value.
2	K	IR	Input for setting gain.
3	T1	ITR	**T**ime **1**. Input for filter time constant.
4	BAL	IB	**BAL**ance. Input for activation of following.
5	BALREF	IR	**BAL**ance **REF**erence. Input for reference value when following.
6	OHL	IR	**O**utput **H**igh **L**imit. Input for upper limit value.
7	OLL	IR	**O**utput **L**ow **L**imit. Input for lower limit value.
10	O	OR	**O**utput. Output signal.
11	O=HL	OB	**O**utput=**H**igh **L**imit. Output which is set to 1 if the output O is limited to the upper limit value.
12	O= LL	OB	**O**utput=**L**ow **L**imit. Output which is set to 1 if the output O is limited to the lower limit value.
13	ERR	OB	**ERR**or. Output set to 1 if OHL is less than OLL.

Diagrams



Bóde diagram



Step response

Function

The step response in the time plane for a single pole low pass filter is: $O(t)=I(t)\ K\ (1–e^{-t/T1})$. The transfer function for a single pole low pass filter is: $G(s)=K(1/(1+sT1))$. This has been implemented in the FILT-1P element as a recursive algorithm.

Gain, Filter Time and Sampling Time

Certain constants are precalculated to make the execution time of the element as short as possible. The result is stored internally in the element. These constants are recalculated if T1 or K are changed by more than 1/128 of their previous value, or if the sampling time TS is changed. When recalculating, a test is performed to check if T1>2 x TS. If not, T1 is set equal to 2 x TS.

Following

If BAL is set to 1, the filter immediately goes into following and output O is set to the value of input BALREF. If the value at BALREF exceeds the output signal limits, the output is set to the limit value concerned. Return to dynamic state is bumpless.

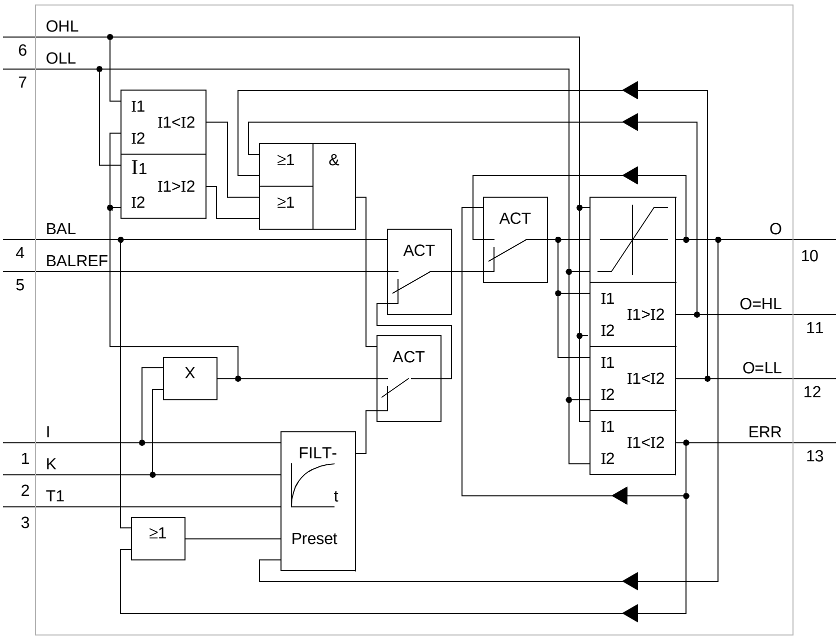
Limitation Function

The limitation function limits the output signal to the limit values at input OHL for upper limit value, and input OLL for lower limit value.

If the actual value exceeds the upper limit value, the output O=HL is set to 1. If it falls below the lower limit value, the output O=LL is set to 1.

When the limitation status has been detected, a check is made each time the element is executed to determine if K x I(t) exceeds the output signal limitations. If so, the limitation status remains. If not, the calculation of the output signal is performed by the algorithm in the normal way. Return from limitation to a dynamic state is bumpless. The element checks that the upper limit value OHL is greater that the lower limit value OLL. If not, the output ERR is set to 1.

While the error status persists, the outputs O=HL, O=LL and O retain the values they had in the sample before, in which the error occurred. After an error, the return to a dynamic state is bumpless, in the same way as in the case above.



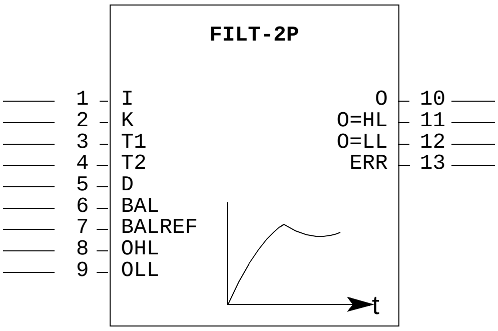
Function diagram

Filter with 2 Poles FILT-2P

Summary

FILT-2P (**FILT**er - **2 P**ole) is used as a 2-pole low pass filter.

The output signal can be limited with limit values specified at special inputs. The balancing function permits the output signal to follow an external reference and permits a bumpless return to the normal function. All transfers from static states are bumpless.



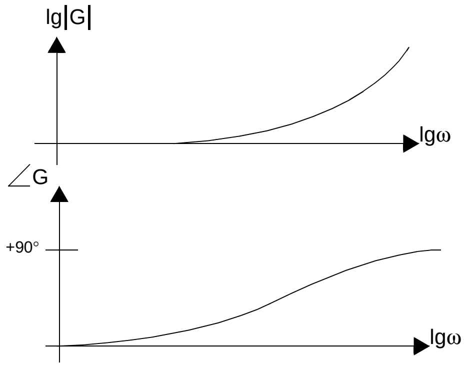
PC element FILT-2P

CALL FILT–2P

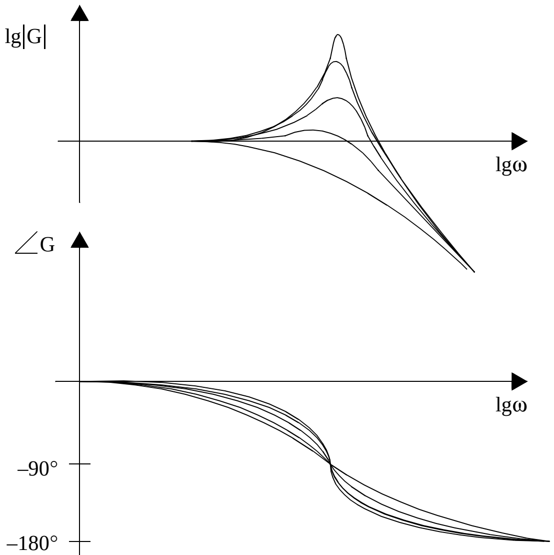
Terminal Description

No	Name	Type	Description
1	I	IR	**I**nput. Input signal.
2	K	IR	Input for setting gain.
3	T1	ITR	**T**ime **1**. Input for 1/resonance angular frequency.
4	T2	ITR	**T**ime **2**. Derivation time constant.
5	D	IR	**D**amping. Damping constant.
6	BAL	IB	**BAL**ance. Input for activation of following.
7	BALREF	IR	**BAL**ance **REF**erence. Input for reference value when following.
8	OHL	IR	**O**utput **H**igh **L**imit. Input for upper limit value.
9	OLL	IR	**O**utput **L**ow **L**imit. Input for lower limit value.
10	O	OR	**O**utput. Output signal.
11	O= HL	OB	**O**utput=**H**igh **L**imit. Output which is set to 1 if the output is limited by the upper limit value.
12	O= LL	OB	**O**utput=**L**ow **L**imit. Output which is set to 1 if the output is limited by the lower limit value.
13	ERR	OB	**ERR**or. Output set to 1 if OHL is less than OLL.

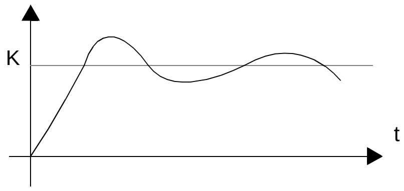
Diagrams



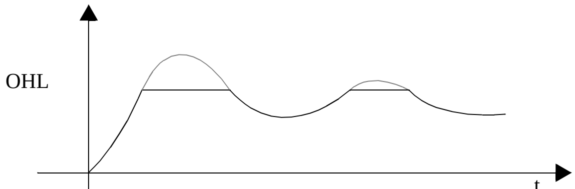
Bode diagram for 1/(sT1)



Bode diagram for 1/(1 x s2DT2 x (sT2)2)



Step response



Function in limiting state

Function

The transfer function for a 2-pole low pass filter is: G(s)=K(1+sT1)/(1+s2DT2+(sT2)^2)

This has been implemented in the FILT-2P element as a recursive algorithm. The design of this algorithm is such that normal functioning is maintained even during limiting. This ensures a controlled return to a dynamic state.

Gain, Filter Time, Damping Factor and Sampling Time

Certain constants are precalculated to make the execution time of the element as short as possible. The results are stored internally in the element. These constants are recalculated if T1, T2, D or K are changed by more than 1/128 of their previous value or if the sampling time TS is changed.

When recalculating, a test is performed to check whether 0<T1<2 x TS. If this is the case, T1 is set equal to 2 x TS.
In the same way a test is made to check if D<0.1, in which case D is set to 0.1. If T2>2 x TS, T2 is set to 2 x TS.

T1 can be set to 0, thus making the element a pure low pass filter.

Following

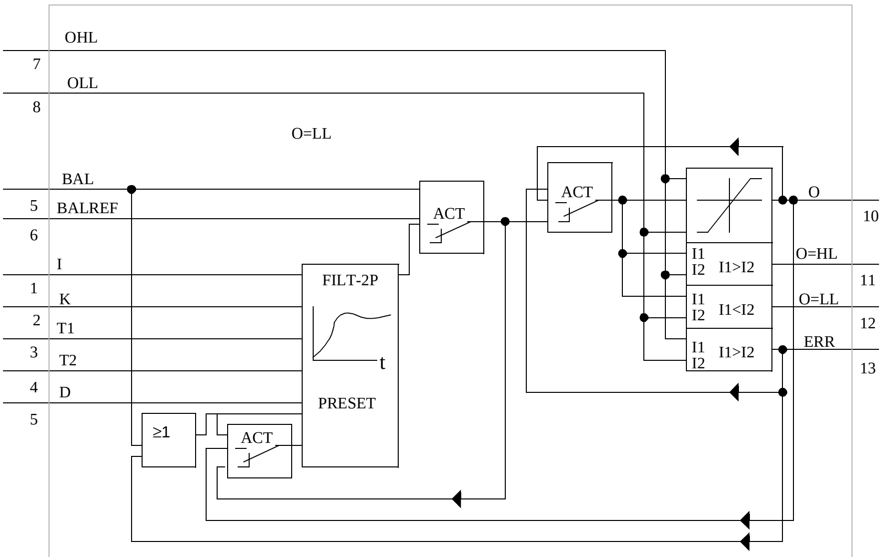
If BAL is set to 1, the filter immediately goes into following and the output O is set to the value of the input BALREF. If the value at BALREF exceeds the output signal limits, the output is set to the limit value concerned. Return to dynamic state is bumpless.

Limitation Function

The limitation function limits the output signal to the limit values at the inputs OHL for upper limit value, and OLL for the lower limit value.

If the actual value exceeds the upper limit value, the output O=HL is set to 1 and if it falls below the lower limit value, the output O=LL is set to 1. The element checks whether the upper limit value OHL is greater than the lower limit value OLL. If not, the output ERR is set to 1.

While the error status persists, the outputs O=HL, O=LL and O retain the values they had in the sample before that in which the error occurred. After an error the return to a dynamic state is bumpless, in the same way as in the case of following above.

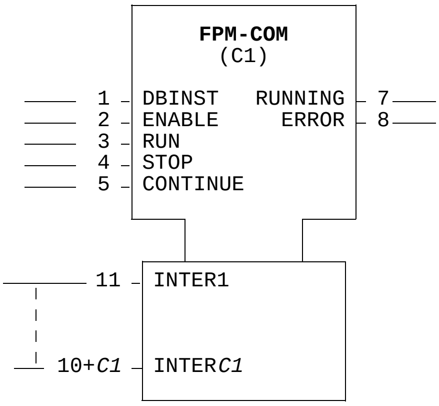


Function diagram

Free-Programmable Module Commands FPM-COM

Summary

FPM-COM (**F**ree-**P**rogrammable **M**odule **COM**mands) is used to control the execution of the application program on the free-programmable module (PU535) and to retrieve status information back from there.



PC element FPM COM

Call FPM-COM

Parameter	Significance	Permissible values
C1	Number of interrupt inputs	0 - 8

Terminal Description

No	Name	Type	Description
1	DBINST	II	An operational parameter or a constant, defining the instance number of the data base element (to be found in its header). Alternatively, a connection to the data base element.
2	ENABLE	IB	Activation of FPM-COM. If set to 1: Control transfer occurs. If set to 0: The old signal values are frozen and no change occurs.
3	RUN	IB	Start execution of the application program at the beginning, if set to 1. If it changes to 0, the current application program cycle stops after completing. A further change from 0 to 1 causes a new start after clearing of all local data.
4	STOP	IB	Stop execution of the application program after completing the program cycle. Further activation is possible by CONTINUE.
5	CONTINUE	IB	Continue execution of the application program from the beginning without clearing of local data.
7	RUNNING	OB	Set to 1, if application program is being executed. The output is set to 0, if the program is terminated via RUN(0), STOP(1) or on error.

Terminal Description

No	Name	Type	Description
8	ERROR	OI	Error codes: If an error occurs concerning the terminal connections of FPM-COM or any runtime error on the FPM, this output holds the corresponding error code. -1: --- (not used by FPM-COM) -2: Bus error -3: --- (not used by FPM-COM) -4: Module not initialized -5: Execution error in application program -6: DBINST input not connected -7: ---(not used by FPM-COM) -8: Incorrect module identification -9: No proper application program available -10: Input time-out on FPM occurred -11: Output time-out on FPM occurred -12: No corresponding interrupt routine -13: Data type error
11	INTER1	IB	If the input is set to 1, the application program stops after ending and a special service routine is started if implemented. If not implemented, an error code is set and the normal application program is continued.
10+*C1*	INTER*C1*	IB	Number of interrupt inputs is given by parameter *C1*.

Function

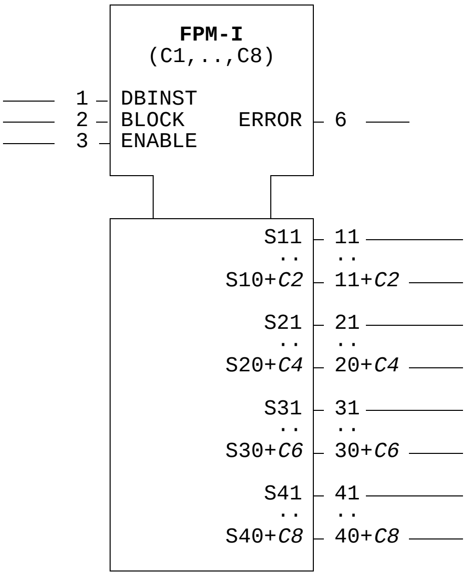
FPM-COM holds the basic address of the module. As a maximum of 10 FPMs can be installed in one Advant Controller 410 or Advant Controller 450, for each of them a separate address is required. Command instructions (start/stop/continue/interrupts) are transferred to the DPM on the FPM and any status informations from there are read to present it on the output terminals.

All control commands transferred on this way are interpreted at FPMs side only at the end of the running application program. If no program is active, then interpretation is done immediately. If an interrupt input is set to 1, then an interrupt service routine is executed on FPM after normally ending the running cycle.

Free-Programmable Module, Input FPM-I

Summary

FPM-I (**F**ree-**P**rogrammable **M**odule - **I**nput) reads data from the free-programmable module (PU535) and puts them on the output terminals Sxx. From there they are available for further processing in the PC program.



PC element FPM-I

Call FPM-I (C1, - , C8)

Parameter	Significance	Permissible values
C1	Data type for section 1	B, I, IL, R, T, TR
C2	Data size for section 1	0 - .5
C3	Data type for section 2	B, I, IL, R, T, TR
C4	Data size for section 2	0 - .5
C5	Data type for section 3	B, I, IL, R, T, TR
C6	Data size for section 3	0 - .5
C7	Data type for section 4	B, I, IL, R, T, TR
C8	Data size for section 4	0 - .5

Terminal Description

No	Name	Type	Description
1	DBINST	II	An operational parameter or a constant, defining the instance number of the data base element (to be found in its header). Alternatively, a connection to the data base element.
2	BLOCK	II	Block number for data in the memory on the FPB. Permitted values are 1 - .22.
3	ENABLE	IB	Activates FPM-I.

Terminal Description

No	Name	Type	Description
6	ERROR	OI	If an error occurs, an error code is presented at this output.
			Error codes: -1:--- (not used by FPM-I) -2:Bus error -3:--- (not used by FPM-I) -4:Module not initialized -5:--- (not used by FPM-I) -6:DBINST input not connected -7:Invalid block number -8:Incorrect module identification
11 - 10+*C2*	S11 - S10+*C2*	O*C1*	Output data of data type in accordance to the parameter *C1*.
21 - 20+*C4*	S21 - S20+*C4*	O*C3*	Output data of data type in accordance to the parameter *C3*.
31 - 30+*C6*	S31 - S30+*C6*	O*C5*	Output data of data type in accordance to the parameter *C5*.
41 - 40+*C8*	S11 - S40+*C8*	O*C7*	Output data of data type in accordance to the parameter *C7*.

Function

FPM-I reads all assigned signals from the module in accordance to the given block address. Data is read only if all data is updated by the FPM,

One FPM-I is able to transfer maximum 20 signals, independently of the signal type. This amount of data is subdivided into 4 sections with 5 signals each. For each section a different size and a different data type may be defined by call parameters. Using up to 22 FPM-I functions, a total number of 440 signals can be transferred theoretically.

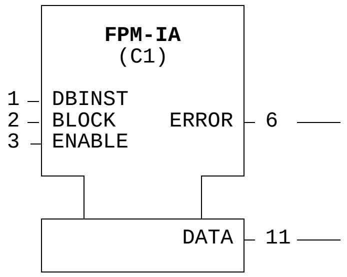
NOTE

The total number of input and output elements for one FPM module is 22.

Free-Programmable Module Input Array FPM-IA

Summary

FPM-IA (**F**ree-**P**rogrammable **M**odule - **I**nput **A**rray) can read a string of 80 characters max from the memory on the free programmable module (PU535). The string is represented as a byte array in the PC element.



PC element FPM-IA

Call FPM -IA (C1)

Parameter	Significance	Permissible values
C1	SIze of byte array	1 - 80

Terminal Description

No	Name	Type	Description
1	DBINST	II	An operational parameter or a constant, defining the instance number of the data base element (to be found in its header). Alternatively, a connection to the data base element.
2	BLOCK	II	Block number for data in the memory. Permitted values are 1 - .22.
3	ENABLE	IB	Activates FPM-IA.
6	ERROR	OI	Error codes: -1: --- (not used by FPM-IA) -2: Bus error -3: --- (not used by FPM-IA) -4: Module not initialized -5: --- (not used by FPM-IA) -6: DBINST input not connected -7: Invalid block number -8: Incorrect module identification
11	DATA	OA*C1*	Output for data

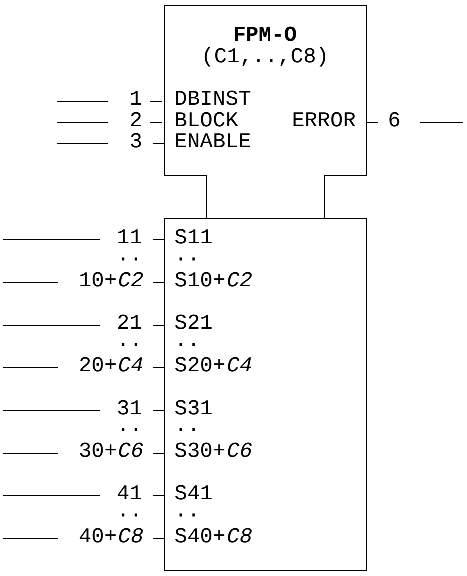
Function

The FPM-IA copies a data string from the FPM. If C1 is smaller than the length of the string, the transferred string is shortened to C1. Equivalent to the mechanism on normal process data transfer, the string data is read only if it is updated since the last time it was read.

Free-Programmable Module, Output FPM-O

Summary

FPM-O (Free-Programmable Module - Output) writes its (input terminal) signals into the free-programmable module (PU535) in accordance to the assigned block number.



PC element FPM-O

Call FPM-O (C1, - ,C8)

Parameter	Significance	Permissible values
C1	Data type for section 1	B, I, IL, R, T, TR
C2	Data size for section 1	0 - .5
C3	Data type for section 2	B, I, IL, R, T, TR
C4	Data size for section 2	0 - .5
C5	Data type for section 3	B, I, IL, R, T, TR
C6	Data size for section 3	0 - .5
C7	Data type for section 4	B, I, IL, R, T, TR
C8	Data size for section 4	0 - .5

Terminal Description

No	Name	Type	Description
1	DBINST	II	An operational parameter or a constant, defining the instance number of the data base element (to be found in its header). Alternatively, a connection to the data base element.
2	BLOCK	II	Block number for data in the memory. Permitted values are 1 - .22.

Terminal Description

No	Name	Type	Description
3	ENABLE	IB	Activates FPM-O.
6	ERROR	OI	If an error occurs, an error code is presented at this output. Error codes: -1: --- (not used by FPM-O) -2: Bus error -3: --- (not used by FPM-O) -4: Module not initialized -5: --- (not used by FPM-O) -6: DBINST input not connected -7: Invalid block number -8: Incorrect module identification
11 - 10+*C2*	S11 - S10+*C2*	I*C1*	Input data of data type in accordance to the parameter *C1*.
21 - 20+*C4*	S21 - S20+*C4*	I*C3*	Input data of data type in accordance to the parameter *C3*.
31 - 30+*C6*	S31 - S30+*C6*	I*C5*	Input data of data type in accordance to the parameter *C5*.
41 - 40+*C8*	S11 - S40+*C8*	I*C7*	Input data of data type in accordance to the parameter *C7*.

Function

FPM-O reads data from its input terminals and writes them out into the DPM on the FPM (only if the last written data is read by the FPM). From there data are available for reading by the FPM application program (as input data).

One FPM-O element is able to transfer a maximum of 20 signals, independently of the signal type. This amount of data is subdivided into 4 sections with 5 signals each. For each section a different size and a different data type may be defined by call parameters.

As maximum 20 signals can be handled by one FPM-O function, a total number of 440 signals can be transferred theoretically, using 22 FPM-O functions.

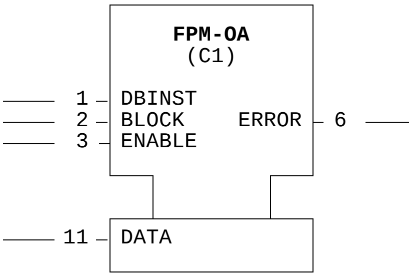
NOTE

The total number of input and output elements for one FPM module is 22.

Free-Programmable Module, Output Array FPM-OA

Summary

FPM-OA (**F**ree-**P**rogrammable **B**oard - **O**utput **A**rray) can write a byte array of 80 characters max into the memory on the free-programmable module (PU535).



PC element FPM-OA

Call FPM-OA(C1)

Parameter	Significance	Permissible values
C1	Size of Array	1 - 80

Terminal Description

No	Name	Type	Description
1	DBINST	II	An operational parameter or a constant, defining the instance number of the data base element (to be found in its header). Alternatively, a connection to the data base element.
2	BLOCK	II	Block number for data in the memory. Permitted values are 1 - .22.
3	ENABLE	IB	Activates FPM-OA.
6	ERROR	OI	If an error occurs, an error code is presented at this output. Error codes: -1:--- (not used by FPM-OA) -2:Bus error -3:--- (not used by FPM-OA) -4:Module not initialized -5:--- (not used by FPM-OA) -6:DBINST input not connected -7:Invalid block number -8:Incorrect module identification
11	DATA	IA*C1*	Input for data.

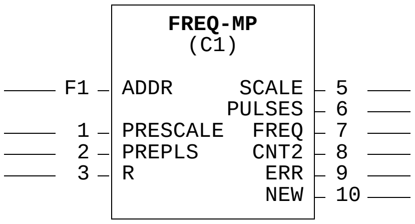
Function

The FPM-OA copies an array to the FPM. New data is only transferred, if the old data is read by the FPM.

Frequency Counter FREQ-MP

Summary

FREQ-MP (FREQuency - Multiple Pulses) is used with DSDP 150 for measuring frequencies from 20 Hz - 10 kHz.



PC element FREQ-MP

Call FPM-OA(C1)

Parameter	Significance	Permissible values
C1	Indicates the number of the counter channel.	1 - 6

Terminal Description

No	Name	Type	Description
F1	ADDR	II	**ADDR**ess. Strapped I/O address on DSDP 150 in decimal code. The board must be strapped within the interval for Misc Board which is address 192 to 239.
1	PRESCALE	II	**PRESCALE**. Input for start value of SCALE. If PRESCALE is set to a negative value, this value is forced to SCALE.
2	PREPLS	II	**PRE**set **P**u**LS**es. Input for pre setting the counter for the process signalzCH1.
3	R	IB	**R**eset. Input which clears the counter and forces the value of SCALE to that at the input PRESCALE. With measurement of pulse rations R must be constantly 1.
5	SCALE	OI	Optimized value for clock frequency division factor.
6	PULSES	OI	Optimized value for number of measurement periods.
7	FREQ	OR	**FREQ**uency. Output for frequency in Hz.
8	CNT2	OIL	Output for value from Counter 2.
9	ERR	OB	**ERR**or. Output for error signal which indicates that the element has made 5 consecutive calls to the board without receiving a value. An error signal is also given when an addressing error occurs on the DSDP 150.
10	NEW	OB	**NEW** value. Output which is set to 1 during one execution cycle each time a new value has been read.

Function

FREQ-MP is used for either frequency measurement or measuring pulse ratios. When a measured value has been presented, the next measurement starts no earlier than one period later, depending on the next execution of the PC element.

Frequency Measurement

For this measurement two counters are used. Counter 1 counts a preset number of cycles representing the time during which the measurement is to be done. Counter 2 counts pulses from the clock (500 kHz) divided by SCALE.

During clearing with the input R, SCALE is held at the value at the input PRESCALE and the value PULSES is held equal to PREPLS. If the count value CNT2, when read, is outside the interval 4000 - 10000 the value of SCALE will be changed at the next measurement.

After initialization of the system, PRESCALE and PREPLS are used as the values for SCALE and PULSES to get an approximate value at the first measurement. The frequency division is required to avoid exceeding the maximum value of the counter. The frequency (FREQ) is calculated according to:

$$FREQ = \frac{PULSES \times 500000}{CNT2 \times SCALE}$$

Pulse Ratio Measurement

Counter 1 is loaded with the value at PREPLS after which the counter is stepped down with pulses from the associated process input 1.

Counter 2 counts pulses associated with input 2.

When counter 1 reaches 0, the counting for counter 2 is interrupted and the value (CNT2) is made available. The pulse ratio may then be calculated with external PC elements according to:

$$\frac{CNT2}{PREPLS}$$

Strapping of DSDP 150

Each pair of counters on the DSDP 150 that is used with FREQ-MP must have strap group S1XX where XX is the counter’s number strapped according to the following.

With frequency measurement:

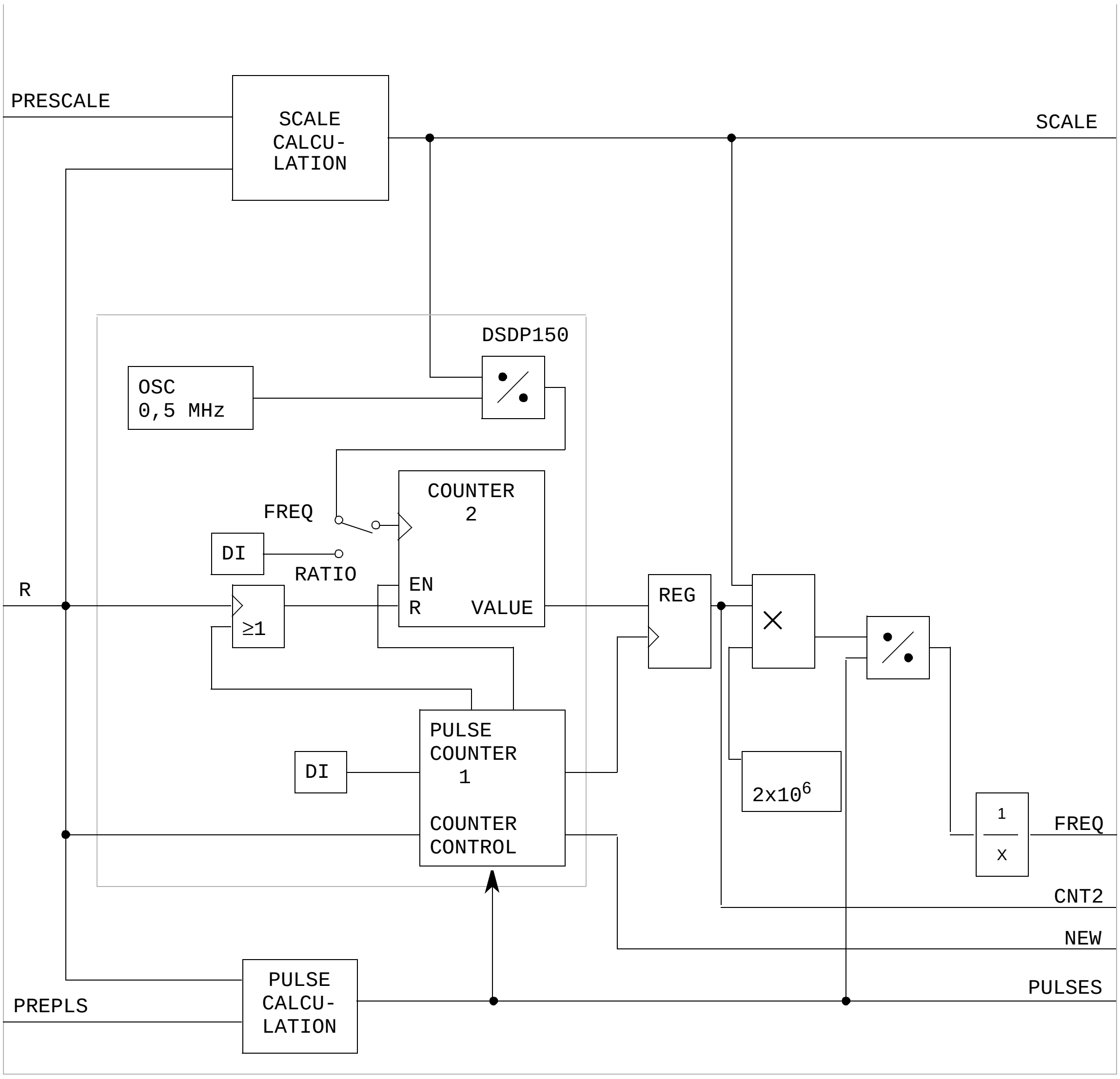
1:st counter: 1-2 and 7-8
2:nd counter: 2-3 and 7-10.

With pulse ratio measurement:

1:st counter: 1-2 and 7-8
2:nd counter 1-2 and 7-10.

Error Supervision

If counter 2 counts more pulses than can be represented in 16 bits (65535), no measured value is obtained, and the output ERR is set to 1. The element must then be reset. This is performed with the R input. The value at the input PREPLS must then be decreased. An error signal is also obtained when an addressing error occurs for DSDP 150.

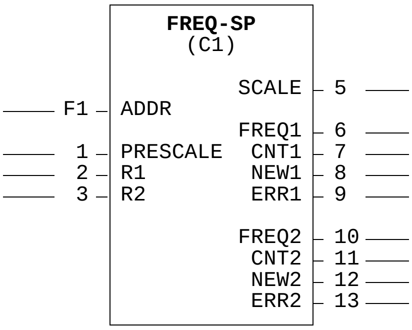


Function diagram

Low Frequency Counter FREQ-SP

Summary

FREQ-SP (**FREQ**uency **-** **S**ingle **P**ulse) is used with DSDP 150 for measurement of frequencies up to 1 kHz.



PC element FREQ-SP

Call FREQ-SP (C1)

Parameter	Significance	Permissible values
C1	Gives the group number of the counter channels. A group consists of two channels.	1 - 6

Terminal Description

No	Name	Type	Description
F1	ADDR	II	**ADDR**ess. Strapped I/O address on DSDP 150 in decimal code. The board must be strapped within the interval for Misc Board which is address 192 to 239.
1	PRESCALE	II	**PRESCALE**. Input for start value of SCALE. If PRESCALE is set to a negative value, this value is forced to SCALE.
2	R1	IB	**R**eset **1**. Input that clears counter 1 and for forcing SCALE to the value of PRESCALE.
3	R2	IB	**R**eset **2**. Input that clears counter 2 and forcing SCALE to the value of PRESCALE.
5	SCALE	OI	Optimized value for division factor for clock frequency.
6	FREQ1	OR	**FREQ**uency **1**. Output for frequency in Hz.
7	CNT1	OIL	**C**ou**NT**er **1**. Output for value read from counter 1.
8	NEW1	OB	**NEW** value for counter **1**. Output which is set to 1 during one execution cycle when a new frequency value has been calculated.
9	ERR1	OB	**ERR**or **1**. Output for error signal which indicates that the element has called the board 100 times without obtaining a measurement value. An error signal is also obtained when an addressing error occurs for the DSDP 150.
10	FREQ2	OR	**FREQ**uency **2**. Output for frequency in Hz.
11	CNT2	OIL	**C**ou**NT**er **2**. Output for value read from counter 2.
12	NEW2	OB	**NEW** value for counter **2**. Output which is set to 1 during one execution cycle when a new frequency value has been calculated.
13	ERR2	OB	**ERR**or **2**. Output for error signal which indicates the element has called the board 100 times without obtaining a measurement value. An error signal is also obtained when an addressing error occurs for the DSDP 150.

Function

FREQ-SP is used with DSDP 150 for measuring low frequencies for two channels. The element may also be used for accurate time measurement.

Frequency Measurement

Frequency measurement is done indirectly by measuring the period time on the relevant digital input. During one full period from the process, an internal clock with frequency 0.5 MHz divided by SCALE is counted. After initialization of the Advant Controller 400 system PRESCALE is used as the value for SCALE to obtain an approximate value at the first measurement.

The frequency division is required to avoid exceeding the maximum value of the counter and to achieve, by way of optimization, the highest possible accuracy in the measurement. If the counted value is less than 2000, the value of SCALE is divided by 2. If the counted value is between 20000 and 30000, the value of SCALE is increased by 3. If the counted value is greater than 30000, the value of SCALE is increased by 10. Such a change is done after each measurement in order to achieve a faster optimization. The output signals are calculated according to:

$$FREQ1 = \frac{500000}{CNT1 \times SCALE}$$

The value in counter 1 is stored at output CNT1 and the value in counter 2 is stored on output CNT2. When a new value has been calculated this is indicated on the outputs VAL2 and NEW2 respectively. When a measurement value has been presented the next measurement starts with the next pulse, that is, the frequency is measured on every other pulse from the process.

Time Measurement

Time measurement of an entire period may be obtained in the same manner as with frequency measurement. The cycle time may be calculated by inverting FREQ by means of external PC elements, or by evaluating:

$$T1 = \frac{SCALE \times CNT1}{500000}$$

Strapping of DSDP 150

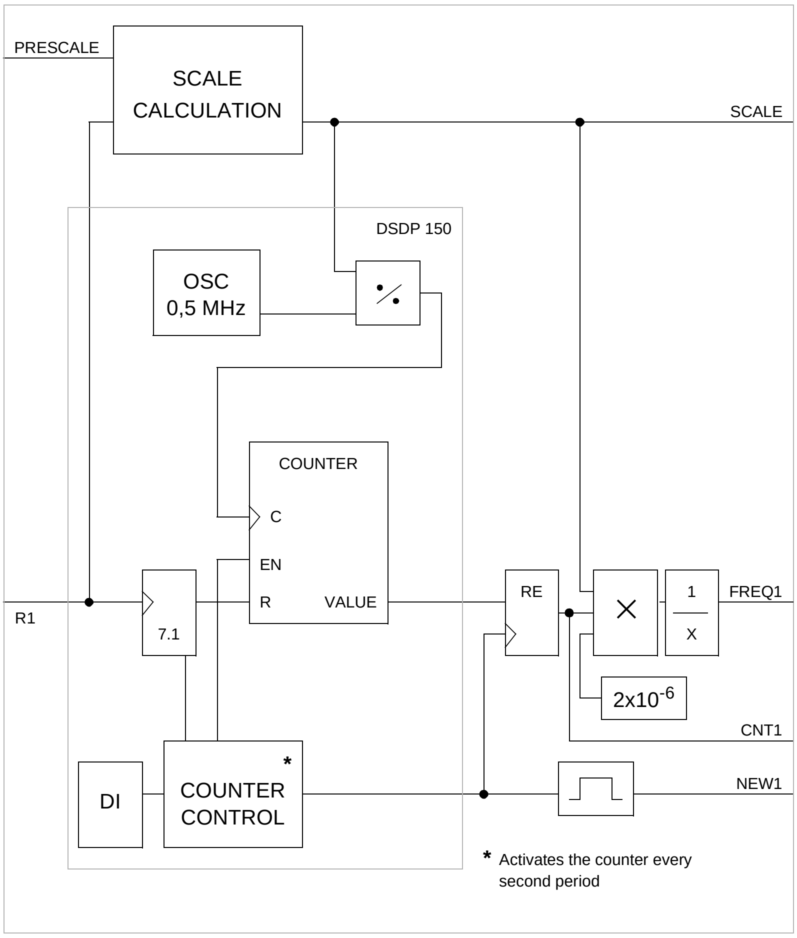
Each counter on the DSDP 150 that is used with FREQ-MP must have strap group S1XX (where XX is the number of the counter) strapped as follows: 2-3 and 4-5.

Error Supervision

If the counter counts more pulses than can be presented by 16 bits (65535), the module automatically restarts the counter. If not, a proper value is presented from the counter, the output ERR is set after:

1 sec if SCALE is less than 6
3 sec if SCALE is between 6 and 40
10 sec if CALE is greater than 40.

When output ERR is set, the output FREQ is set to zero. Output ERR is also set when addressing error occurs for DSDP 150.



Block diagram

Function Header FUNCM

Summary

FUNCM (**FUNC**tion **M**odule) is used for structuring a PC program.

PC Element FUNCM

Call FUNCM

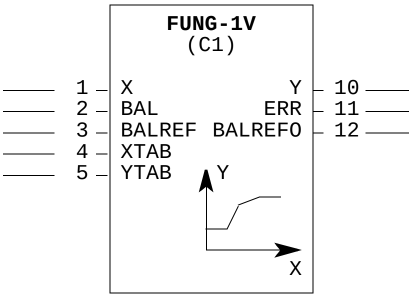
Function

The purpose of FUNCM is to control the arrangement of the documentation. The element cannot affect the execution.
A function module can be used to provide a subdivision of a PC program, a control module, a slave or a sequence step in several functionally associated parts. A function module can contain a control module, slave sequence or a new function module.

Function Generator with 1 Variable FUNG-1V

Summary

FUNG-1V (**FUN**ction **G**enerator - **1** **V**ariable) is used for generation of an optional function of one variable, y=f(x). The function is described by a number of co-ordinates. Linear interpolation is used for values between these coordinates. Maximum 255 coordinates can be specified.



PC element FUNG-1V

Call FUNG-1V (C1)

Parameter	Significance	Permissible values
C1	Number of coordinates which describe the function.	2 - 255

Terminal Description

No	Name	Type	Description
1	X	IR	**X**-value. Input for X-value.
2	BAL	IB	**BAL**ance. Input for activation of balancing.
3	BALREF	IR	**BAL**ance **REF**erence. Value which the Y-output is to adopt with balancing.
4	XTAB	IG*C1*R	**X TAB**le. Group data for the X-table with *C1* values.
5	YTAB	IG*C1*R	**Y TAB**le. Group data for the Y-table with *C1* values.
10	Y	OR	**Y**-value. Output for Y-value.
11	ERR	OB	**ERR**or. Error signal which is set (to 1) if X is outside the value of XTAB or if Y, on balancing, is outside the YTAB values.
12	BALREFO	OR	**BAL**ance **REF**erence **O**utput. Output for calculated X-value with balancing.

Function

FUNG-1V calculates an output signal Y for a value at the input X. The calculation is performed in accordance with a piece by piece linear function which is determined by the vectors XTAB and YTAB. For each X-value in XTAB, there is a corresponding Y-value in YTAB. The Y-value at the output is calculated by means of linear interpolation between the two X-values in XTAB which are nearest the value at the input X. The values in X-tab must be strictly increasing from low to high serial numbers in the table.

Interpolation

The function generated can be illustrated by the following figure. The interpolation is performed as follows:

$y=y_k+(x-x_k)\ (y_{k+1}-y_k)/(x_{k+1}-x_k)$

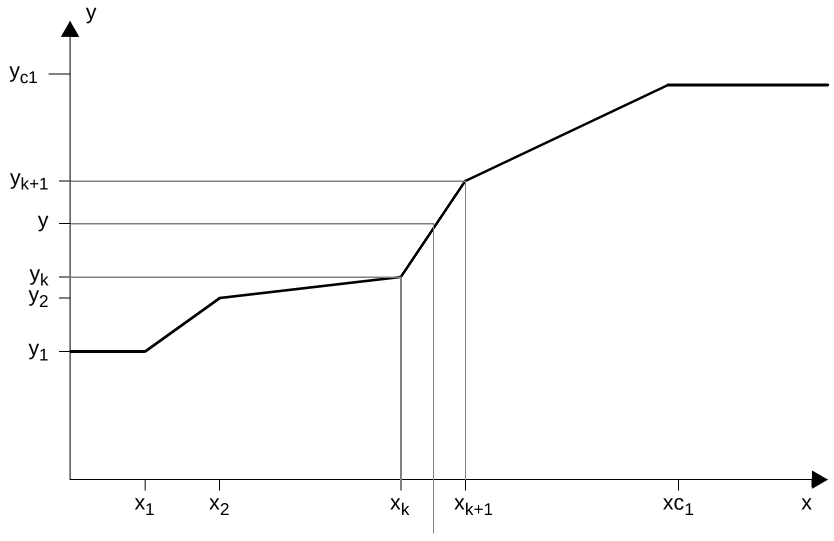
Balancing

On activation of the balancing input BAL, the value at Y is set to the value at the input BALREF. The X-value which corresponds to this Y-value is obtained at the output BALREFO. On balancing, the X-value is calculated by interpolation in the same way as the Y-value is calculated during normal operation. To permit the balancing, the values in YTAB must be strictly increasing or decreasing from low to high serial numbers in the table.

Error signal

If the input signal X is outside the range defined by XTAB, the ERR output is set to 1. The Y-value is then set to the greatest or lowest value resp. in YTAB.

ERR is also set to 1 if BALREF is equal to or outside the YTAB value range when BAL is set to 1. The value at Y is then set to the value at the input BALREF and BALREFO is set to the greatest or lowest value resp. in XTAB.

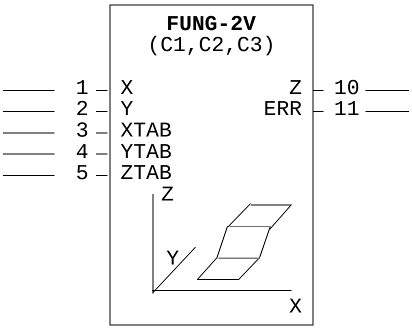


Example of function

Function Generator with 2 Variables FUNG-2V

Summary

FUNG-2V (**FUN**ction **G**enerator **- 2 V**ariables) is used for generation of an optional function of two variables, z=f(x, y).



PC element FUNG-2V

Call FUNG-2V (C1, C2, C3)

Parameter	Significance	Permissible values
C1	Number of X coordinates which describe the function.	2 - 127
C2	Number of Y coordinates which describe the function.	2 - 127
C3	Number of Z values which describe the function.	4 - 255
$2 \le C3 = C1 \mathrm{x} C2 \le 255$		

Terminal Description

No	Name	Type	Description
1	X	R	**X**-value. Input for X value.
2	Y	IR	**Y**-value. Input for Y value.
3	XTAB	IGR*C1*	**X TAB**le. Group data for the X table with *C1* values
4	YTAB	IGR*C2*	**Y TAB**le. Group data for the Y table with *C2* values.
5	ZTAB	IGR(*C1*x*C2)*	**Z TAB**le. Group data for the Z table with *C1*x*C2* values.
10	Z	OR	**Z**-value. Output for Z value.
11	ERR	OB	Error signal which is set (to 1) when X and/or Y is outside XTABs or YTABs values respectively.

Function

FUNG-2V gives an output signal Z which is calculated using linear interpolation for two variables, the values of which are given at the inputs X and Y. The values in XTAB and YTAB must be strictly increasing from low to high serial numbers. For each pair of values in XTAB, YTAB there is one value in ZTAB.

Interpolation

The function generated can be illustrated with the following figure. Interpolation is done according to:

$Z=f(x, y)=f(x_i, y_j)+$
$+K_x\ [f(x_{i+1}, y_j)–f(x_i, y_j)]+$
$+K_y\ [f(x_i, y_{j+1})–f(x_i, y_j)]+$
$+K_xK_y\ [f(x_i, y_j)+f(x_{i+1}, y_{j+1})–f(x_{i+1}, y_j)–f(x_i, y_{j+1})]$

where

$K_x=(x–x_i)/(x_{i+1}–x_i)$
$K_y=(y–y_j)/(y_{j+1}–y_j)$

Group Data

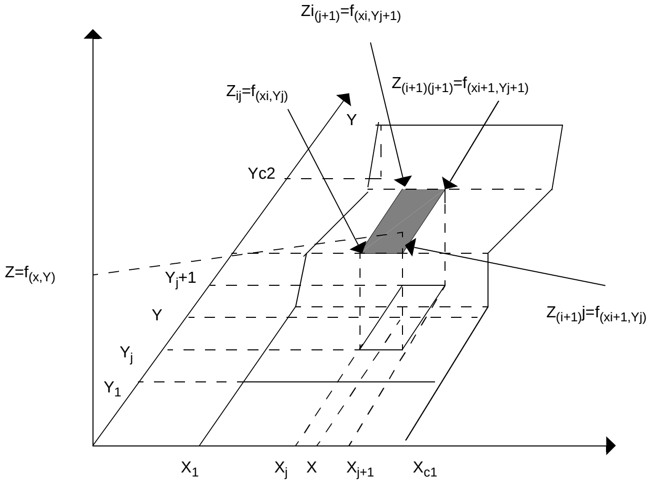
Group data for XTAB, YTAB and ZTAB are stored in accordance with the following:

XTAB: $X_1, X_2 ...X_i... X_{c1}$
YTAB: $Y_z, Y_2 ...Y_j... Y_{c2}$
ZTAB: $Z_{11}, Z_{12} ...Z_{1j}... Z_{1c2}$
$Z_{21}, Z_{22} ...Z_{2j}...Z_{2c2}$
$\vdots$
$Z_{c11}, Z_{c12} ...Z_{c1j}... Z_{c1c2}$

Error Signal

If X and/or Y values are given, which are outside the values in XTAB and YTAB respectively, the error signal ERR is set (to 1) and the output signal is limited to the Z value that applies to the min. or max X and/or Y values in the tables.

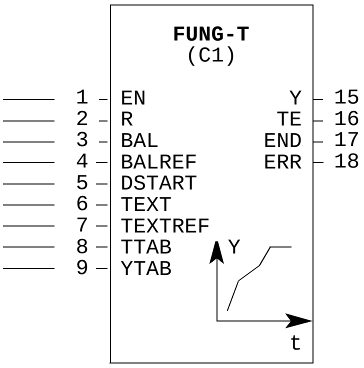
Example of function


$Zi_{(j+1)}=f_{(xi,Yj+1)}$
$Z_{ij}=f_{(xi,Yj)}$
$Z_{(i+1)(j+1)}=f_{(xi+1,Yj+1)}$
Y
Yc2
$Z=f_{(x,Y)}$
Y_j+1
Y
$Z_{(i+1)}j=f_{(xi+1,Yj)}$
Y_j
Y_1
X_1
X_j
X
X_{j+1}
X_{c1}

Function Generator with Time Variable FUNG-T

Summary

FUNG-T (**FUN**ction **G**enerator **- T**ime) is used for generating a signal, the value of which varies with time in accordance with an optional function. It is used in applications in which a process signal is to be controlled in accordance with a predetermined sequence (profile) specified from a start time. The balancing function makes available the possibility of controlling the output signal to track an external reference.



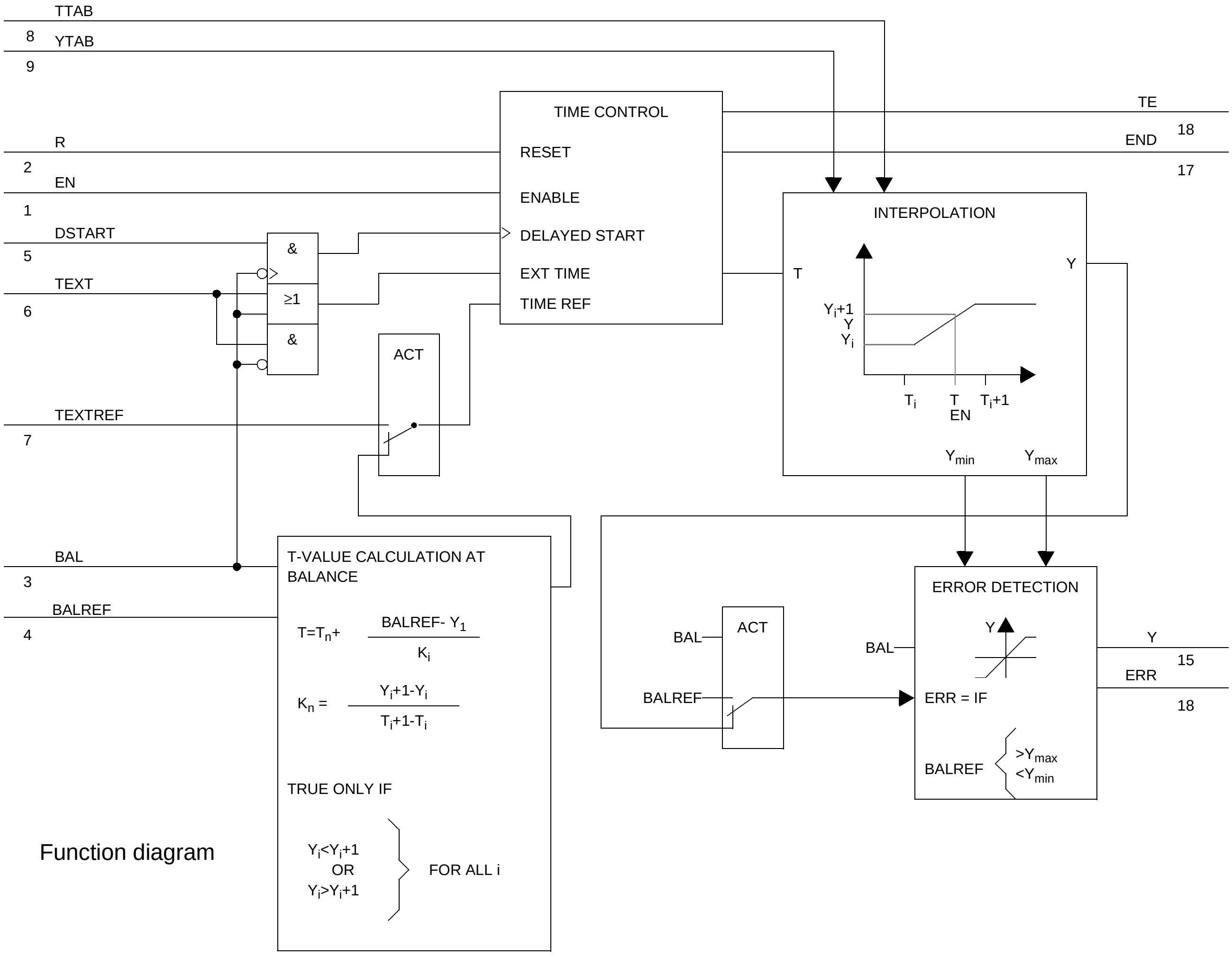
PC element FUNG-T

Call FUNG-T (C1)

Parameter	Significance	Permissible values
C1	The number of break points in the function, that is, the number of places in group data for TTAB and YTAB	2 - 255

Terminal Description

No	Name	Type	Description
1	EN	IB	**EN**able. Input which starts and stops the internal timing. When EN=0, the timing is stopped and the output signal is frozen.
2	RR	IB	**R**eset. TE (time which has expired) is cleared when R is set. R dominates over other inputs.
3	BAL	IB	**BAL**ance. Input for activation of tracking. The output signal follows BALREF and the time for the corresponding point on the time curve is calculated.
4	BALREF	IR	**BAL**ance **REF**erence. The input for reference value when tracking.
5	DSTART	IB	**D**elayed **START**. Input which indicates that change from the tracking is to take place with a delayed start. When BAL goes to 0 and EN=1, the internal timing starts from 0 but the output signal and time value TE remain at the values they had before BAL went to 0 until the internal time reaches the value at TE.
6	TEXT	IB	**T**ime **EXT**ernal is a control signal which indicates that an external time reference is available at the input TEXTREF.
7	TEXTREF	ITR	**T**ime **EXT**ernal **REF**erence- Input for external time reference. If the control signal TEXT is set to 1 the value at this input is used as a time reference.
8	TTAB	IG*C1*TR	**T**ime **TAB**le. Group data input for table values for the time coordinate in the coordinate pair (T_i, Y_i).
9	YTAB	IG*C1*R	**Y TAB**le. Group data input for table values for the Y-coordinate in the coordinate pair (T_i, Y_i).
15	Y	OR	**Y**-value. Output for Y-value.
16	TE	OTR	**T**ime **E**lapsed. Output which indicates the time which has expired, that is, the current position along the time axis.
17	END	OB	Output which indicates that the time has reached the final value (T_{C1}, Y_{C1}).
18	ERR	OB	**ERR**or. Output which is set if $Y>Y_{max}$ or $Y<Y_{min}$ is tracking.



Function diagram

Function

FUNG-T gives an output signal Y which varies with time in accordance with a piecewise linear function with a maximum of 255 break points. The curve which the output signal is to follow is normally determined by coordinate pairs (xxx) - (xxx) stored on the group data inputs TTAB and YTAB. Y can adopt both positive and negative values. When EN is set, the internal timing commences. The output signal TE indicates the time that has expired with EN set. The output END indicates that the time has reached the final value in accordance with the last coordinate pair in the table. The value of Y is calculated with linear interpolation between two adjacent T-values from TTAB according to:

$Y=Y_i+K(T–T_i)$, where $K=(Y_{i+1}–Y_i)/(T_{i+1}zT_i)$

The values in TTAB must be strictly increasing from low to high serial numbers. The end position points on the curve determine the limits of the output signal, that is, for $T<T_1$ becomes $Y=Y_1$ and for $T>T_{C1}$ becomes $Y=Y_{C1}$.

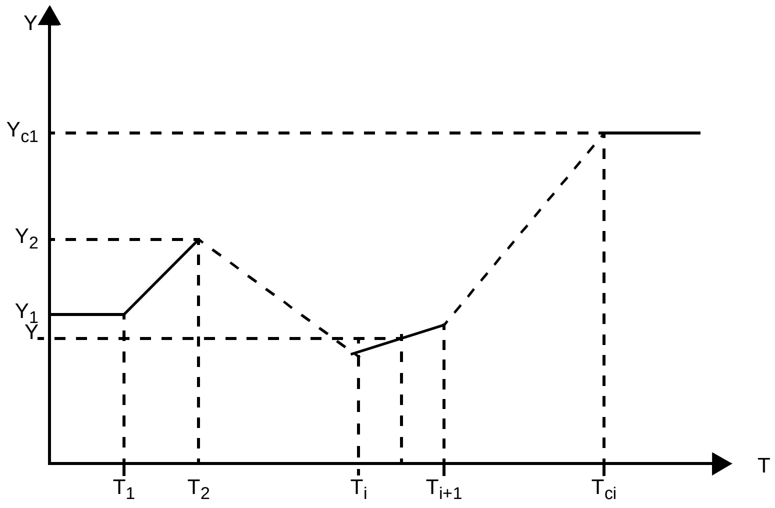
Clearing

When the input R is set, the internal time is cleared and the outputs adopt the values:

$Y=Y_1$ (Start value of the time curve); TE = 0; END = 0; ERR = 0

Tracking

If the control signal BAL (**BAL**ance) is set, FUNG-T goes into tracking. The output Y adopts the value on the input BALREF. In addition the point on the time axis which corresponds to this value is calculated. The result is stored at the output TE.
If tracking is to be used, the curve must be strictly increasing or decreasing. BAL overrides all inputs except R (**R**eset) and gives Y=BALREF; $TE=f^{-1}$(BALREF); END=0; If BALREF>Y_{max} or BALREF<Y_{min} ERR is set.



Example of time curve

Delayed Start

The input signal DSTART gives a delayed start of the calculation of the output signal after tracking. If DSTART=1 when BAL goes to 1, the output signal Y and the time TE retain the values they had at changeover. The internal timing starts from 0 (if EN=1) and when this time reaches TE, the output signal begins to follow the curve and TE continues counting. If En=0, the internal timing stops and the delay is extended. DSTART must be set continuously until the internal time=TE. If DSTART goes to 0 before, the normal function starts from the current output signal. BAL overrides DSTART.

External Time Reference

The internal timing can be replaced by an external signal for time. Selection of external timing is only possible with normal calculation of the output signal and not with tracking or a delayed start.

Supervision and Limitation

With normal function:

- If TE≤T1 Y is set=Y1
- If TE≥TC1 Y is set=YC1 and END is set.

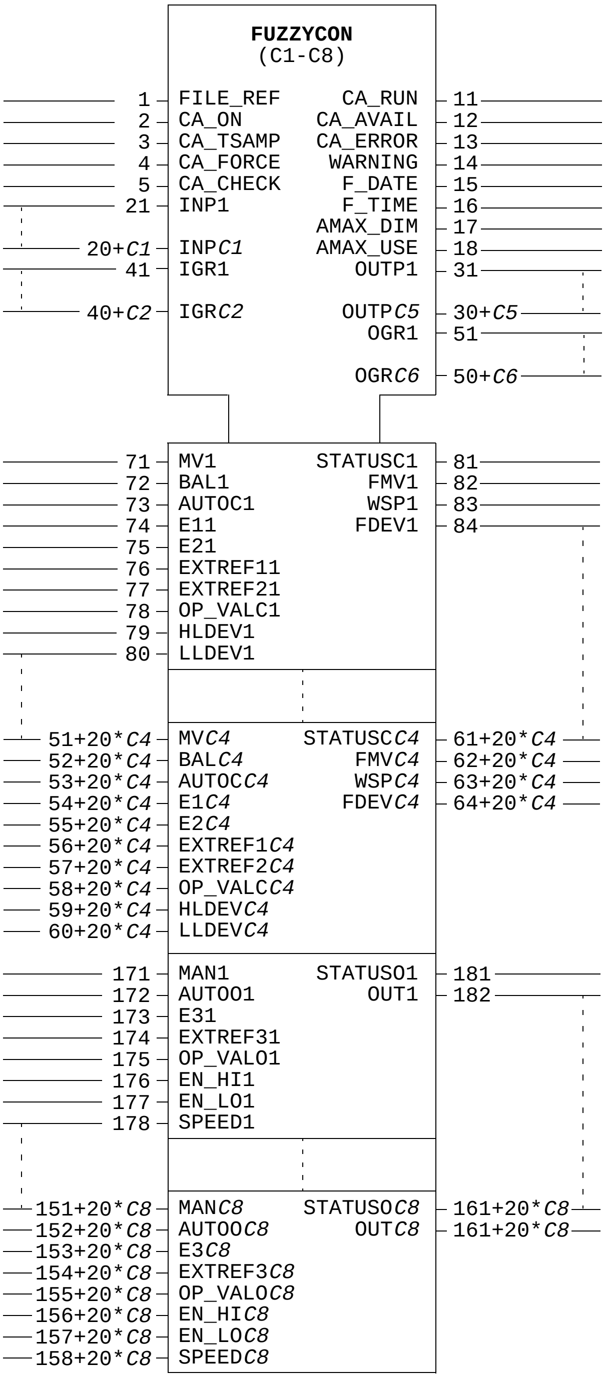
With following:

- If Y>Ymax TE is set=fz1(Ymax) and ERR is set.
- If Y<Ymin TE is set=fz1(Ymin) and ERR is set.

Fuzzy Controller FUZZYCON

Summary

FUZZYCON is used as a **FUZZY CON**troller together with the FTC, Fuzzy Target Code, which is stored in the DB element AAB_FILE. Further information can be found in the user's guide, Fuzzy Control with FUZZYCON.



PC element FUZZYCON

Call FUZZYCON(C1,C2,C3,C4,C5,C6,C7,C8)

Parameter	Significance	Permissible values
C1	Number of auxiliary inputs of type IR.	0 - 5
C2	Number of grouped inputs of type IGR.	0 - 5(1)
C3	Size (number elements) of grouped input data.	0 - 255(1)
C4	Number of controlled variables.	0 - 3
C5	Number of auxiliary outputs of type OR.	0 - 5
C6	Number of grouped outputs of type OGR.	0 - 5(2)
C7	Size (number elements) of grouped output data.	0 - 255(2)
C8	Number of output variables.	0 - 3

(1) C2 > 0 requires that C3 > 0
(2) C6 > 0 requires that C7 > 0

Terminal Description

No	Name	Type	Description
1	FILE_REF	IA20	**FILE REF**erence to the “NAME” property of the AAB_FILE instance. Empty string (D = ‘ ‘) shall be specified in case no file should be referenced.
2	CA_ON	IB	**C**ontrol **A**lgorithm **ON**. FUZZYCON ordered to execute its control algorithm.
3	CA_TSAMP	ITR	**C**ontrol **A**lgorithm **T**ime **SAMP**le. The time interval between the interpretations of the control algorithm, measured in the unit seconds. Rounded off to closest time that is realizable by the CONTRM.
4	CA_FORCE	IB	**C**ontrol **A**lgorithm **FORCE**. Forces interpretation of the control algorithm.
5	CA_CHECK	IB	**C**ontrol **A**lgorithm **CHECK**. The PC element checks the control algorithm for consistency in every execution that interprets the control algorithm.
11	CA_RUN	OB	**C**ontrol **A**lgorithm **RUN**. Set when FUZZYCON has accepted the order CA_ON, which means that the control algorithm will be interpreted when conditions for this is fulfilled.
12	CA_AVAIL	OB	**C**ontrol **A**lgorithm **AVAIL**able. Set when a control algorithm is available in the referenced AAB_FILE.
13	CA_ERROR	OI	**C**ontrol **A**lgorithm **ERROR**. Indicates error regarding the FTC. See Table CA_ERROR at page 198.
14	WARNING	OI	**WARNING**s. Indicates warnings. See Table WARNING at page 199.
15	F_DATE	OIL	**F**ile **DATE** of file that is accepted and interpretable.
16	F_TIME	OTR	**F**ile **TIME** of file that is accepted and interpretable.
17	AMAX_DIM	OIL	**A**llocated **MAX**imum break points **DIM**ensioned. Number of term break points that the compiler has allocated in the work space of FTC.
18	AMAX_USE	OIL	**A**llocated **MAX**imum break points **USE**d. Number of term break points that have been used “so far” in the work space of FTC.
21	INP1	IR	**INP**ut **1**. Auxiliary input of type REAL.

Terminal Description

No	Name	Type	Description
: 20+*C1*	INP*C1*	IR	**INP**ut ***C1***. Auxiliary input of type REAL.
31	OUTP1	OR	**OUTP**ut **1**. Auxiliary output of type REAL.
: 30+*C5*	OUTP*C5*	OR	**OUTP**ut ***C5***. Auxiliary output of type REAL.
41	IGR1	IG*C3*R	**I**nput **GR**oup **1**. Grouped auxiliary input of type REAL.
: 40+*C2*	IGR*C2*	IG*C3*R	**I**nput **GR**oup ***C2***. Grouped auxiliary input of type REAL.
51	OGR1	OG*C7*R	**O**utput **GR**oup **1**. Grouped auxiliary output of type REAL.
: 50+*C6*	OGR*C6*	OG*C7*R	**O**utput **GR**oup ***C6***. Grouped auxiliary output of type REAL.
71	MV1	IR	**M**easured **V**alue of controlled variable **1**.
72	BAL1	IB	**BAL**anced mode for controlled variable **1**.
73	AUTOC1	IB	**AUTO** mode for **C**ontrolled variable **1**.
74	E11	IB	**E1** mode for controlled variable **1**.
75	E21	IB	**E2** mode for controlled variable **1**.
76	EXTREF11	IR	**EXT**ernal **REF**erence value **1** (set point) for controlled variable **1** during mode E1. Internally, the value is limited by the limits for the working set point.
77	EXTREF21	IR	**EXT**ernal **REF**erence value **2** (set point) for controlled variable **1** during mode E2. Internally, the value is limited by the limits for the working set point.
78	OP_VALC1	IG7R	**OP**erator **VAL**ues for **C**ontrolled variable **1**. A group of REAL data, normally coming from operator via GENCON. See Table OP_VALC1 - OP_VALCC4 at page 199.
79	HLDEV1	IR	**H**igh alarm **L**imit for **DEV**iation between filtered measured value and working set point for controlled variable **1**.
80	LLDEV1	IR	**L**ow alarm **L**imit for **DEV**iation between filtered measured value and working set point for controlled variable **1**.
81	STATUSC1	OI	**STATUS** for **C**ontrolled variable **1**. BOOLEAN data, normally delivered to operator station via GENCON. See Table STATUSC1 - STATUSCC4 at page 199.
82	FMV1	OR	**F**iltered **M**easured **V**alue for controlled variable **1**.
83	WSP1	OR	**W**orking **S**et **P**oint for controlled variable **1**, i.e. “SETP” (see Table OP_VALC1 - OP_VALCC4 at page 199) in mode AUTO, “EXTREF1” in mode E1 and “EXTREF2” in mode E2. This value is updated only when the control algorithm is executed.
84	FDEV1	OR	**F**iltered **DEV**iation for controlled variable **1**, i.e. filtered measured value minus working set point.
: 51+20**C4*	MV*C4*	IR	**M**easured **V**alue of controlled variable ***C4***.

Terminal Description

No	Name	Type	Description
: 52+20**C4*	BAL*C4*	IB	**BAL**anced mode for controlled variable ***C4***.
: 53+20**C4*	AUTOC*C4*	IB	**AUTO** mode for **C**ontrolled variable ***C4***.
: 54+20**C4*	E1*C4*	IB	**E1** mode for controlled variable ***C4***.
: 55+20**C4*	E2*C4*	IB	**E2** mode for controlled variable ***C4***.
: 56+20**C4*	EXTREF1*C4*	IR	**EXT**ernal **REF**erence value **1** (set point) for controlled variable ***C4*** during mode E1. Internally, the value is limited by the limits for the working set point.
: 57+20**C4*	EXTREF2*C4*	IR	**EXT**ernal **REF**erence value **2** (set point) for controlled variable ***C4*** during mode E2. Internally, the value is limited by the limits for the working set point.
: 58+20**C4*	OP_VALC*C4*	IG7R	**OP**erator **VAL**ues for **C**ontrolled variable ***C4***. A group of REAL data, normally coming from operator via GENCON. See Table OP_VALC1 - OP_VALCC4 at page 199.
: 59+20**C4*	HLDEV*C4*	IR	**H**igh alarm **L**imit for **DEV**iation between filtered measured value and working set point for controlled variable ***C4***.
: 60+20**C4*	LLDEV*C4*	IR	**L**ow alarm **L**imit for **DEV**iation between filtered measured value and working set point for controlled variable ***C4***.
: 61+20**C4*	STATUSC*C4*	OI	**STATUS** for **C**ontrolled variable ***C4***. BOOLEAN data, normally delivered to operator station via GENCON. See Table STATUSC1 - STATUSCC4 at page 199.
: 62+20**C4*	FMV*C4*	OR	**F**iltered **M**easured **V**alue for controlled variable ***C4***.
: 63+20**C4*	WSP*C4*	OR	**W**orking **S**et **P**oint for controlled variable ***C4***, i.e. “SETP” (see Table OP_VALC1 - OP_VALCC4 at page 199) in mode AUTO, “EXTREF1” in mode E1 and “EXTREF2” in mode E2. This value is updated only when the control algorithm is executed.
: 64+20**C4*	FDEV*C4*	OR	**F**iltered **DEV**iation for controlled variable ***C4***, i.e. filtered measured value minus working set point.
171	MAN1	IB	**MAN**ual mode for output variable **1**.
172	AUTOO1	IB	**AUTO** mode for **O**utput variable **1**.
173	E31	IB	**E3** mode for output variable **1**.
174	EXTREF31	IR	**EXT**ernal **REF**erence value **3** for output variable **1**. Applied output in mode E3.
175	OP_VALO1	IG3R	**OP**erator **VAL**ues for **O**utput variable **1**. A group of REAL data, normally coming from operator via GENCON or AOC. See Table OP_VALO1 - OP_VALOC8 at page 200.
176	EN_HI1	IB	**EN**ables the limit “**HI**_LIM” (see Table OP_VALO1 - OP_VALOC8 at page 200) for output variable **1**.
177	EN_LO1	IB	**EN**ables the limit “**LO**_LIM” (see Table OP_VALO1 - OP_VALOC8 at page 200) for output variable **1**.

Terminal Description

No	Name	Type	Description
178	SPEED1	IR	**SPEED** limit of output value during mode AUTO and E3 for output variable **1**, measured per second.
181	STATUSO1	OI	**STATUS** for **O**utput variable **1**. BOOLEAN data. For use within AMPL and/or delivered to operator by GENCON or AOC. See Table STATUSO1 - STATUSC8 at page 200.
182	OUT1	OR	**OUT**put value for output variable **1**.
: 151+20**C8*	MAN*C8*	IB	**MAN**ual mode for output variable ***C8***.
: 152+20**C8*	AUTOO*C8*	IB	**AUTO** mode for **O**utput variable ***C8***.
: 153+20**C8*	E3*C8*	IB	**E3** mode for output variable ***C8***.
: 154+20**C8*	EXTREF3*C8*	IR	**EXT**ernal **REF**erence value **3** for output variable ***C8***. Applied output in mode E3.
: 155+20**C8*	OP_VALO*C8*	IG3R	**OP**erator **VAL**ues for **O**utput variable ***C8***. A group of REAL data, normally coming from operator via GENCON or AOC. See Table OP_VALO1 - OP_VALOC8 at page 200.
: 156+20**C8*	EN_HI*C8*	IB	**EN**ables the limit “**HI**_LIM” (see Table OP_VALO1 - OP_VALOC8 at page 200) for output variable ***C8***.
: 157+20**C8*	EN_LO*C8*	IB	**EN**ables the limit “**LO**_LIM” (see Table OP_VALO1 - OP_VALOC8 at page 200) for output variable ***C8***.
: 158+20**C8*	SPEED*C8*	IR	**SPEED** limit of output value during mode AUTO and E3 for output variable ***C8***, measured per second.
: 161+20**C8*	STATUSO*C8*	OI	**STATUS** for **O**utput variable ***C8***. BOOLEAN data. For use within AMPL and/or delivered to operator by GENCON or AOC. See Table STATUSO1 - STATUSC8 at page 200.
: 162+20**C8*	OUT*C8*	OR	**OUT**put value for output variable ***C8***.

CA_ERROR(1)

Error	Description
1	One or both check sum values are wrong. Probably caused by transmission error when loading FTC from AS100ES to database.
2	`signatureA` is wrong. Probable cause. Probably caused by transmission error when loading FTC from AS100ES to database.
3	`targetSystem` is wrong. Probably caused by wrong argument for compiler option *-t* during compilation.
4	`targetFunction` is wrong. Probable cause. Probably caused by transmission error when loading FTC from AS100ES to database.

CA_ERROR[(1)]

Error	Description
5	`fTCid` is wrong. Indicated when the version of AFPLCompiler and PC Element mismatch.
6	`memoryModel` is wrong. Probable cause. Probably caused by transmission error when loading FTC from AS100ES to database.
7	One or several call parameters of FTC do not match the call parameters of the PC element. Probable cause is a mismatch made in the source code of the control algorithm, or wrong specification of call parameters
8	AAB_FILE instance not found.
9	No AAB_FILE referenced and CA_ON set to true.
10	Unidentified instruction within interpreted code encountered (implies mismatch between versions of AFPLCompiler and interpreter).

(1) Indicates erroneous properties of FTC

WARNING

Bit	Description
1(1)	At least one controlled variable in BAL mode.
2	All output variables in non automatic modes (MAN or E3).

(1) Least Significant Bit.

OP_VALC1 - OP_VALC*C4*

Parameter	Description
1	“SETP”. Applied reference value (set point) during mode AUTO. Internally the value is limited to the limits for the working set point.
2	“MVH2”. High alarm limit 2 for filtered measured value.
3	“MVH1”. High alarm limit 1 for filtered measured value.
4	“MVL1”. Low alarm limit 1 for filtered measured value.
5	“MVL2”. Low alarm limit 2 for filtered measured value.
6	“SETPH”. High limit for working set point.
7	“SETPL”. Low limit for working set point.

STATUSC1 - STATUSC*C4*

Bit	Description
1(1)	“BAL”. Set when mode is BAL.
2	“AUTO”. Set when mode is AUTO.
3	“E1”. Set when mode is E1.

STATUSC1 - STATUSC*C4*

Bit	Description
4	“E2”. Set when mode is E2.
5	“ALDEV”. Set when deviation is outside “LLDEV” … “HLDEV”.
6	“ALLIMH2”. Set when filtered measured value is larger than “MVH2”.
7	“ALLIMH1”. Set when filtered measured value is larger than “MVH1”.
8	“ALLIML1”. Set when filtered measured value is smaller than “MVL1”.
9	“ALLIML2”. Set when filtered measured value is smaller than “MVL2”.
10	“LIMSPH”. Set when working set point is limited by “SETPH”.
11	“LIMSPL”. Set when working set point is limited by “SETPL”.

(1) Least Significant Bit.

OP_VALO1 - OP_VALO*C8*

Parameter	Description
1	“VALUE”. Applied output value during mode MAN.
2	“HI_LIM”. High limit for output value.
3	“LO_LIM”. Low limit for output value.

STATUSO1 - STATUS*C8*

Bit	Description
1(1)	“MAN”. Set when mode is MAN.
2	“AUTO”. Set when mode is AUTO.
3	“E3”. Set when mode is E3.
4	“LIMOPH”. Set when output is limited by “HI_LIM”.
5	“LIMOPL”. Set when output is limited by “LO_LIM”.
6	“LIMOSPE”. Set when speed of output is limited by “SPEED”. No indication supported by GENCON or AOC, and speed limit is applied only in mode AUTO and E3.
7	“EN_HI”. Set when FUZZYCON has enabled the high limit.
8	“EN_LO”. Set when FUZZYCON has enabled the low limit.

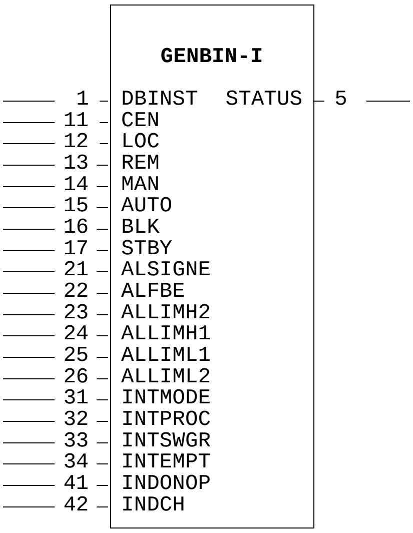
(1) Least Significant Bit.

Binary Functional Unit, Inputs GENBIN-I

Summary

GENBIN-I (**GEN**eral **BIN**ary controller **- I**nputs) is used as an interface for binary signals between the user's PC program and the data base element. The element forms part of the functional unit GENBIN. Further information can be found in the manual *Functional Units part 5, GENCON, GENBIN, GENUSD, GENNOV.*

Call GENBIN-I



PC element GENBIN-I

Terminal Description

No	Name	Type	Description
1	DBINST	II	**D**ata **B**ase **INST**ance. An operational parameter or a constant, defining the instance number of the data base element (to be defined in its header). Alternatively, a connection to the data base element.
5	STATUS	OIL	Connection to STATUS on the data base element for writing the values at the indication inputs.
11	CEN	IB	Operator position indication **CEN**tral. Indication of operator position.
12	LOC	IB	Operator position indication **LOC**al. Indication of operator position.
13	REM	IB	Operator position indication **REM**ote. Indication of operator position.
14	MAN	IB	Operation mode indication **MAN**ual. Indication of operation mode.
15	AUTO	IB	Operation mode indication **AUTO**. Indication of operation mode.
16	BLK	IB	Operation mode indication **BL**oc**K**ed. Indication of operation mode.
17	STBY	IB	Operation mode indication **ST**and **BY**. Indication of operation mode.
21	ALSIGNE	IB	**AL**arm indication, **SIGN**al **E**rror. Output for indication of signal error.
22	ALFBE	IB	**AL**arm indication, **F**eed**B**ack **E**rror. Output for indication of feedback error.
23	ALLIMH2	IB	**AL**arm **LIM**it **H**igh **2**. The measured value has exceeded upper alarm limit 2.
24	ALLIMH1	IB	**AL**arm **LIM**it **H**igh **1**. The measured value has exceeded upper alarm limit 1.

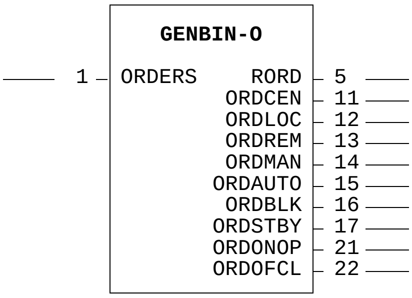
Terminal Description

No	Name	Type	Description
25	ALLIML1	IB	**AL**arm **LIM**it Low **1**. The measured value has fallen blow lower alarm limit 1.
26	ALLIML2	IB	**AL**arm **LIM**it **L**ow **2**. The measured value has fallen below lower alarm limit 2.
31	INTMODE	IB	**INT**erlock **MODE**. Indication of operation mode interlock.
32	INTPROC	IB	**INT**erlock **PROC**ess. Indication of process Interlock.
33	INTSWGR	IB	**INT**erlock **SW**itCH **G**ea**R**. Indication of switch- gear interlock.
34	INTEMTP	IB	**INT**erlock **EM**ergency **T**ri**P**. Indication of emergency trip interlock.
41	INDONOP	IB	**IND**ication **ON/OP**en. Indication of on/open.
42	INDCH	IB	**IND**ication **CH**anging. Indication of change in progress.

Binary Functional Unit, Outputs GENBIN-O

Summary

GENBIN-O (**GEN**eral **BIN**ary controller - **O**utputs) is used as an interface for binary signals between Advant Station 500 Series OS and the user's PC program. The element forms part of the functional unit GENBIN. Further information can be found in the manual *Functional Units part 5, GENCON, GENBIN, GENUSD, GENNOV.*



PC element GENBIN-O

Call GENBIN-O

Terminal Description

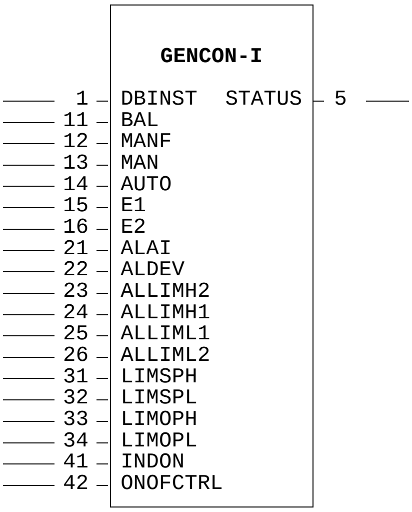
No	Name	Type	Description
1	ORDERS	IIL	Connection to ORDERS on the data base element for reading the values at the order outputs.
5	RORD	OI	**R**eset **ORD**er. Connection to RORD on the data base element to obtain pulsed outputs.
11	ORDCEN	OB	**ORD**er output operator position **CEN**tral. Ordering of central operator position.
12	ORDLOC	OB	**ORD**er output operation position **LOC**al. Ordering of local operator position.
13	ORDREM	OB	**ORD**er output operator position **REM**ote. Ordering of remote control.
14	ORDMAN	OB	**ORD**er output **MAN**ual mode. Ordering of manual operation.
15	ORDAUTO	OB	**ORD**er output **AUTO** mode. Ordering of automatic operation.
16	ORDBLK	OB	**ORD**er output **BL**oc**K**ed mode. Ordering of blocking.
17	ORDSTBY	OB	**ORD**er output **ST**and **BY** mode. Ordering of standby.
21	ORDONOP	OB	**ORD**er output **ON**/**OP**en. On-order.
22	ORDOFCL	OB	**ORD**er output **OF**f/**CL**ose. Off-order.

Analog Functional Unit, Inputs GENCON-I

Summary

GENCON-I (**GEN**eral **CON**troller **- I**nputs) is used as an interface for binary signals between the user's PC program and Advant Station 500 Series OS via the data base element. The element forms part of the functional unit GENCON. Further information can be found in the manual *Functional Units part 5, GENCON, GENBIN, GENUSD, GENNOV.*

Call GENCON-I



PC element GENCON-I

Terminal Description

No	Name	Type	Description
1	DBINST	II	**D**ata **B**ase **INST**ance. An operational parameter or a constant, defining the instance number of the data base element (to be defined in its header). Alternatively, a connection to the data base element.
5	STATUS	OIL	Connection to STATUS on the data base element for writing the values at the indication inputs.
11	BAL	IB	Operation mode indication **BAL**anced. Indication of balanced operation mode.
12	MANF	IB	Operation mode indication **MAN**ual **F**orced. Indication of manual forced control.
13	MAN	IB	Operation mode indication **MAN**ual. Indication of manual operation mode.
14	AUTO	IB	Operation mode indication **AUTO**. Indication of operation mode auto.
15	E1	IB	Operation mode indication **E1**. Indication of operation mode E1.
16	E2	IB	Operation mode indication **E2**. Indication of operation mode E2.
21	ALAI	IB	**AL**arm **A**nalog **I**nput. Indication of error on analog input.
22	ALDEV	IB	**AL**arm **DEV**iation. Control deviation outside the permitted range.
23	ALLIMH2	IB	**AL**arm **LIM**it **H**igh **2**. The measured value has exceeded upper alarm limit 2.
24	ALLIMH1	IB	**AL**arm **LIM**it **H**igh **1**. The measured value has exceeded upper alarm limit 1.
25	ALLIML1	IB	**AL**arm **LIM**it **L**ow **1**. The measured value has fallen below lower alarm limit 1.

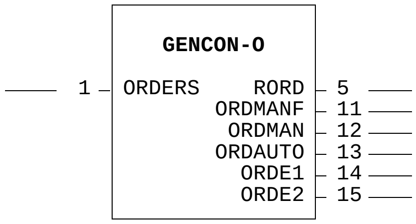
Terminal Description

No	Name	Type	Description
26	ALLIML2	IB	**AL**arm **LIM**it **L**ow **2**. The measured value has fallen below lower alarm limit 2.
31	LIMSPH	IB	**LIM**it **S**et **P**oint **H**igh. The set point has been limited to a high limit value.
32	LIMSPL	IB	**LIM**it **S**et **P**oint **L**ow. The set point has been limited to a low limit value.
33	LIMOPH	IB	**LIM**it **O**ut**P**ut **H**igh. The output signal has been limited to a high limit value.
34	LIMOPL	IB	**LIM**it **O**ut**P**ut **L**ow. The output signal has been limited to a low limit value.
41	INDON	IB	**IND**ication **ON**. If ONOFCTRL is set to 1, INDON indicates ON if it is set to 1 and OFF if it is set to 0.
42	ONOFCTRL	IB	**ON**/**OF**f **C**on**TR**o**L**. If this input is set to 1, ON/OFF indication is obtained.

Analog Functional Unit, Outputs GENCON-O

Summary

GENCON-O (**GEN**eral **CON**troller **- O**utputs) is used as an interface for binary signals between Advant Station 500 Series OS and the user's PC program via the data base element. The element forms part of the functional unit GENCON. Further information can be found in the manual *Functional Units part 5, GENCON, GENBIN, GENUSD, GENNOV.*



PC element GENCON-O

Call GENCON-O

Terminal Description

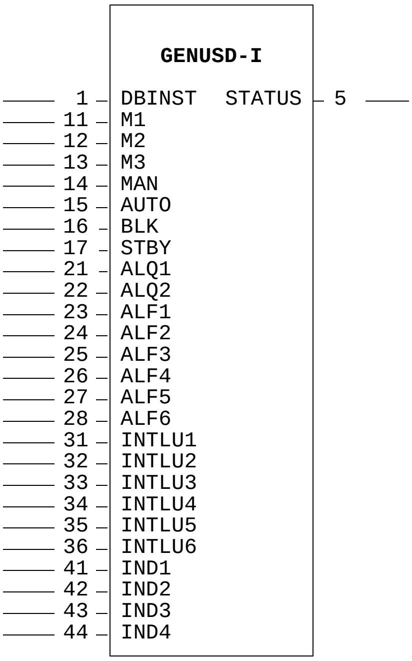
No	Name	Type	Description
1	ORDERS	IIL	Connection to ORDERS on the data base element for reading the values at the order outputs.
5	RORD	OI	**R**eset **ORD**er. Connection to RORD on the data base element to obtain pulsed outputs.
11	ORDMANF	OB	**ORD**er output **MAN**ual **F**orced mode. Ordering of manual forced control.
12	ORDMAN	OB	**ORD**er output **MAN**ual mode. Ordering of manual operation.
13	ORDAUTO	OB	**ORD**er output **AUTO** mode. Ordering of automatic operation.
14	ORDE1	OB	**ORD**er output **AUTO** mode. Ordering of automatic operation.
15	ORDE2	OB	**ORD**er output **E2** mode. Ordering of operation mode E2.

User Defined Functional Unit, Inputs GENUSD-I

Summary

GENUSD-I (**GEN**eral **US**er **D**efined functional unit **- I**nputs) is used as an interface for binary signals between the user's PC program and the Advant Station 500 Series OS via data base element. The element forms part of the functional unit GENUSD. Further information can be found in the manual *Functional Units part 5, GENCON, GENBIN, GENUSD, GENNOV.*

Call - GENUSD-I



PC element GENUSD-I

Terminal Description

No	Name	Type	Description
1	DBINST	II	**D**ata **B**ase **INST**ance. An operational parameter or a constant, defining the instance number of the data base element (to be defined in its header). Alternatively, a connection to the data base element.
5	STATUS	OIL	Connection to STATUS on the data base element for writing the values at the indication inputs.
11	M1	IB	Operator position indication **M1**. Indication of operator position M1.
12	M2	IB	Operator position indication **M2**. Indication of operator position M2.
13	M3	IB	Operator position indication **M3**. Indication of operator position M3.
14	MAN	IB	Operation mode indication **MAN**ual. Indication of manual operation.
15	AUTO	IB	Operation mode indication **AUTO**. Indication of automatic operation.
16	BLK	IB	Operation mode indication **BL**oc**K**ed. Indication of blocking.
17	STBY	IB	Operation mode indication **ST**and **BY**. Indication of standby.
21	ALQ1	IB	**AL**arm indication **Q1**. Alarm indication, signal fault 1.

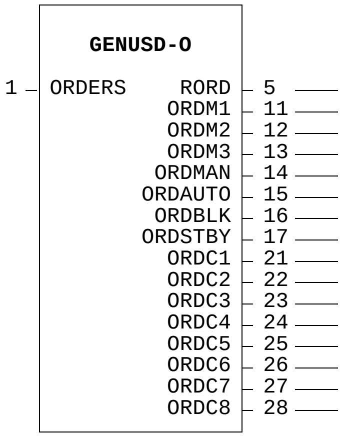
Terminal Description

No	Name	Type	Description
22	ALQ2	IB	**AL**arm indication **Q2**. Alarm indication, signal fault 2.
23	ALF1	IB	**AL**arm indication **F1**. Alarm indication, fault 1.
24	ALF2	IB	**AL**arm indication **F2**. Alarm indication, fault 2.
25	ALF3	IB	**AL**arm indication **F3**. Alarm indication, fault 3.
26	ALF4	IB	**AL**arm indication **F4**. Alarm indication, fault 4.
27	ALF5	IB	**AL**arm indication **F5**. Alarm indication, fault 5.
28	ALF6	IB	**AL**arm indication **F6**. Alarm indication, fault 6.
31	INTLU1	IB	**INT**er**L**ock indication **U1**. Indication of interlock U1.
32	INTLU2	IB	**INT**er**L**ock indication **U2**. Indication of interlock U2.
33	INTLU3	IB	**INT**er**L**ock indication **U3**. Indication of interlock U3.
35	INTLU5	IB	**INT**er**L**ock indication **U5**. Indication of interlock U5.
34	INTLU4	IB	**INT**er**L**ock indication **U4**. Indication of interlock U4.
36	INTLU6	IB	**INT**er**L**ock indication **U6**. Indication of interlock U6.
41	IND1	IB	**IND**ication **1**
42	IND2	IB	**IND**ication **2**
43	IND3	IB	**IND**ication **3**
44	IND4	IB	**IND**ication **4**

User Defined Functional Unit, Outputs GENUSD-O

Summary

GENUSD-O (**GEN**eral **US**er **D**efined functional unit **-** **O**utputs) is used as an interface for binary signals between Advant Station 500 OS and the user's PC program via the data base element. The element forms part of the functional unit GENUSD. Further information can be found in the manual *Functional Units, Part 5, GENCON, GENBIN, GENUSD, GENNOV.*



PC element GENUSD-O

Call GENUSD-O

Terminal Description

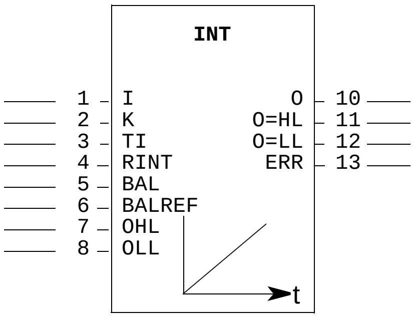
No	Name	Type	Description
1	ORDERS	IIL	Connection to ORDERS on the data base element for reading the values at the order outputs.
5	RORD	OI	**R**eset **ORD**er. Connection to RORD on the data base element to obtain pulsed outputs.
11	ORDM1	OB	**ORD**er output operator position **M1**. Ordering of operator position M1.
12	ORDM2	OB	**ORD**er output operator position **M2**. Ordering of operator position M2.
13	ORDM3	OB	**ORD**er output operator position **M3**. Ordering of operator position M3.
14	ORDMAN	OB	**ORD**er output **MAN**ual mode. Ordering of manual control.
15	ORDAUTO	OB	**ORD**er output **AUTO** mode. Ordering of automatic operation.
16	ORDBLK	OB	**ORD**er output **BL**oc**K**ed mode. Ordering of blocking.
17	ORDSTBY	OB	**ORD**er output **ST**and **BY** mode. Ordering of standby.
21	ORD*C1*	OB	**ORD**er output ***C1***.
22	ORD*C2*	OB	**ORD**er output ***C2***.
23	ORD*C3*	OB	**ORD**er output ***C3***.
24	ORD*C4*	OB	**ORD**er output ***C4***.
25	ORD*C5*	OB	**ORD**er output ***C5***.
26	ORD*C6*	OB	**ORD**er output ***C6***.
27	ORD*C7*	OB	**ORD**er output ***C7***.
28	ORD*C8*	OB	**ORD**er output ***C8***

Integrator INT

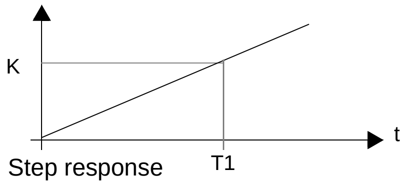
Summary

INT (**INT**egrator) is used to give an integration effect. The output signal can be limited with limit values specified at special inputs. The balancing function permits the output signal to follow an external reference and permits a bumpless return to the normal function.

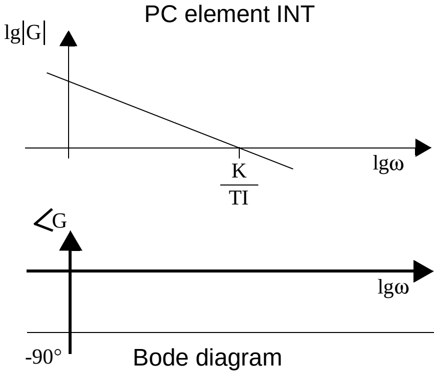
Call INT



PC element INT



Step response



Bode diagram

Terminal Description

No	Name	Type	Description
1	I	IR	**I**nput. Input signal.
2	K	IR	Input for setting of gain.
3	TI	ITR	**T**ime **I**ntegration. Input for time constant for integration.
4	RINT	IB	**R**eset **INT**egrator. Input for clearing of integrator.
5	BAL	IB	**BAL**ance REFerence. Input for reference value when following.
6	BALREF	IR	**BAL**ance **REF**erence. Input for reference value when following.
7	OHL	IR	**O**utput **H**igh **L**imit. Input for upper limit value.
8	OLL	IR	**O**utput **L**ow **L**imit. Input for lower limit value.
10	O	OR	**O**utput. Output signal.
11	O=HL	OB	**O**utput=**H**igh **L**imit. Output which is set to 1 if the output is limited to the upper limit value.
12	O=LL	OB	**O**utput=**L**ow **L**imit. Input which is set to 1 if the output is limited to the lower limit value.
13	ERR	OB	**ERR**or. Output which is set to 1 if OHL is less than OLL.

Function, Transfer Function

The INT function can be written in the time plane as $O(t)=K/TI(\int^{\tau} I(\tau)\, d\, \tau)$. The main property when controlling is that the output signal retains its value when the input signal I(t)=0.The step response in the time plane is O(t)=k x I(t) x t/TI. The transfer function for an integrator is G(s)=K(1/sTI)

Gain, Integration Time Constant and Sampling Time

The constant K x TS/TI is precalculated to reduce the execution time of the element to a minimum and the result is stored internally in the element. This constant is recalculated if TI or K is changed by more than 1/128 of their previous values or if the sampling time TS is changed. When recalculating a test is made to see whether TS/TI<1.TS/TI is otherwise set equal to 1.

Clearing of Integrator

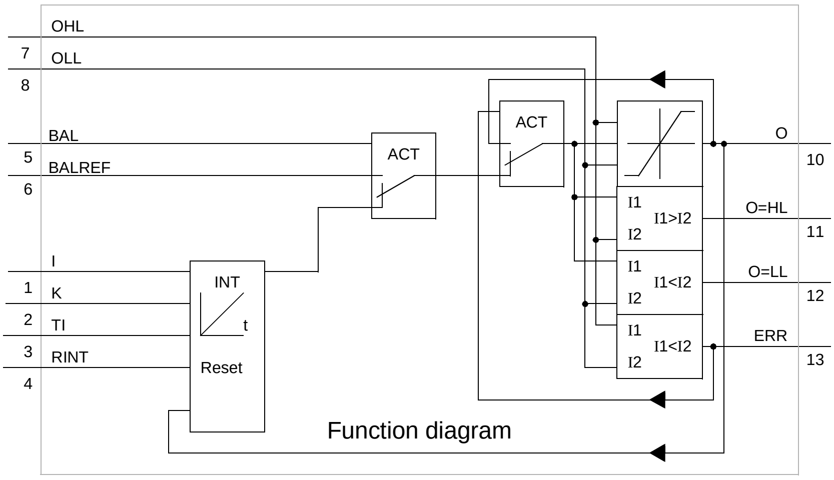
The algorithm is cleared when RINT goes to 1.

Following

If BAL is set to 1, the regulator immediately goes into following and the output O is set to the value of the input BALREF.
If the value at BALREF exceeds the output signal limits, the output is set to the limit value concerned. On return to the normal function, the value of output O during the last sample in following remains a further sample time, after which integration will be performed for this value.

Limitation Function

The limitation function limits the output signal to the values at the inputs OHL for upper limit and OLL for the lower limit. If the actual value exceeds the upper limit, the output O=HL is set to 1 and if it falls below the lower limit, the output O=LL is set to 1. The element checks that the upper limit value OHL is greater than the lower limit value OLL. If this is not the case, the output ERR is set to 1. While the error status persists, the outputs O=HL, O=LL and O retain the values they had in the sample before that in which the error occurred. After limitation or error status, normal integration is performed from the current value.



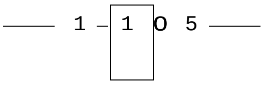
NOTE

AC 410/ 450, revisions 1.0/0 through 1.0/3, is not able to handle increments smaller than $1/ 10^7$ of the output value.

Inverter INV

Summary

INV is used for inverting Boolean variables and particularly when reading signals to the data base.



PC element INV

Call INV

Terminal Description

No	Name	Type	Description
1	--	IB	Input.
5	--	OB	Output of inverted input value.

Function

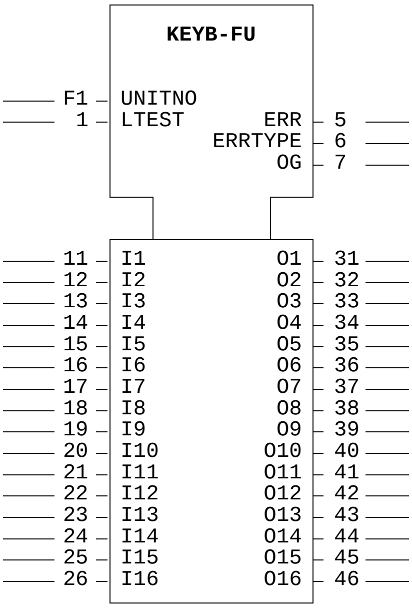
The output signal from the INV element is set (to 1) if the input signal to the element is 0 and is reset (to 0) when the input signal is 1.

Function Keyboard KEYB-FU

Summary

KEYB-FU (**KEYB**oard-**FU**nction) is used for communication with keyboard DSHM 130. The element is used when the keyboard is not divided into groups.

Call KEYB-FU



PC element KEYB-FU

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

Keyboard layout DSHM 130

Terminal Description

No	Name	Type	Description
F1	UNITNO	FI	**UNIT N**umber. Address for MasterView 100 unit. The hundreds and thousands indicate the number of the MasterView 100 bus, the units and tens indicate the number of the node.
1	LTEST	IB	**L**amp **TEST**. Input for activation of lamp test.
5	ERR	OB	**ERR**or. Output for indication of error.
6	ERRTYPE	OI	**ERR**or **TYPE**. Output to determine type of error.
7	OG	OI	**O**utput **G**roup. Indicates number of the key that is pressed. 0 indicates that no key is pressed.
11	I1	IB	**I**nput **1**. Input for lamp in key 1.
12	I2	IB	**I**nput **2**. Input for lamp in key 2.
: 26	I16	IB	**I**nput **16**. Input for lamp in key 16.
31	O1	OB	**O**utput **1**. Output for key 1.
32	O2	OB	**O**utput **2**. Output for key 2.
: 46	O16	OB	**O**utput **16**. Output for key 16.

Function

KEYB-FU is used to permit communication with function keyboard DSHM 130. This element makes it possible to obtain the status of the 16 keys and to control the corresponding lamps in the keys. The keys on the keyboard are numbered as shown in shown in the figure on the previous page. The value at outputs O1 - O16 shows the status of the keys. If an output is set, the corresponding key is pressed.

NOTE

The value of the key only remains at the output for as long as the key is pressed.

The lamps in the keyboard are controlled by the status of inputs I1 - I16. If an input is set, the corresponding lamp is lit. All lamps in the keyboard can be lit by setting the LTEST input, regardless of which keys are pressed.

Error Supervision

When the element indicates an error, the error output ERR is set to 1, and the type of error is indicated at the ERRTYPE output.

The element can indicate the following error types:

0= The element has not detected an error (ERR=0)

1= Controller board to specified MasterView 100 bus missing

2= Controller board to specified MasterView 100 bus faulty

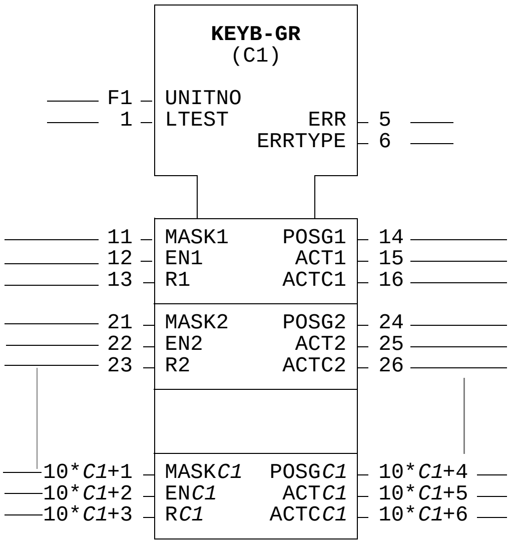
3= The addressed MasterView 100 unit is missing or faulty

15= Illegal value at function parameter UNITNO.

Group Function Keyboard KEYB-GR

Summary

KEYB-GR (**KEYB**oard-**GR**oup).is used for communication with function keyboard DSHM 130 when its keys are divided into groups. The keyboards can be divided up into a maximum of 9 groups with 1 - 16 keys per group.



PC element KEYB-GR

Call KEYB-GR (C1)

Parameter	Significance	Permissible values
C1	Number of groups into which the keyboard is divided.	1 - 9

Terminal Description

No	Name	Type	Description
F1	UNITNO	FI	**UNIT N**umber. Address for MasterView 100 unit. The hundreds and thousands indicate the number of the MasterView 100 bus, the units and tens indicate the number of the node.
1	LTEST	IB	**L**amp **TEST**. Input for activation of lamp test.
5	ERR	OB	**ERR**or. Output for indication of error.
6	ERRTYPE	OI	**ERR**or **TYPE**. Output to determine type of error.
11	MASK1	IIL	**MASK 1**. Input for specification of the inputs included in group 1.
12	EN1	IB	**EN**able **1**. Input for blocking of group tripping or, in the case of group function with one key, lamp control.
13	R1	IB	**R**eset **1**. Input for clearing the group value.
14	POSG1	OI	**POS**ition **G**roup **1**. Output for key number or, in the case of group function with one key, key status.

Terminal Description

No	Name	Type	Description
15	ACT1	OB	**ACT**ive **1**. Output which is set to 1 if POSG1 is other than 0.16ACTC1OB**ACT**ive **C**ycle **1**. Output for reset pulse on key press.
16	ACT*C1*	OB	**ACT**ive **C**ycle **1**. Output for reset pulse on key press.
21	MASK2	IIL	**MASK 2**. Input for specification of the inputs included in group 2.
22	EN2	IB	**EN**able **2**. Input for blocking of group tripping or, in the case of group function with one key lamp control.
23	R2	IB	**R**eset **2**. Input for clearing of group value.
24	POSG2	OI	**POS**ition **G**roup **2**. Output for key number or, in the case of group function with one key, key status.
25	ACT2	OB	**ACT**ive **2**. Output which is set to 1 if POSG2 is other than 0.
26	ACTC2	OB	**ACT**ive **C**ycle **2**. Output for reset pulse on key press.
: 10×*C1*+1	MASK*C1*	IIL	**MASK *C1***. Input for specification of the inputs included in group *C4*.
10×*C1*+2	EN*C1*	IB	**EN**able ***C1***. Input for blocking of group tripping or, in the case of group function with one key, lamp control.
10×*C1*+3	R*C4*	IB	**R**eset ***C1***. Input for clearing of group value.
10×*C1*+4	POSG*C1*	OI	**POS**ition **G**roup ***C1***. Output for key number or, in the case of group function with one key, key status.
10×*C1*+5	ACT*C1*	OB	**ACT**ive ***C1***. Output which is set to 1 if POSGC1 is other than 0.
10×*C1*+6	ACTC*C1*	OB	**ACT**ive **C**ycle ***C1***. Output for reset pulse on key press.

Function

KEYB-GR is used for communication with function keyboard DSHM 130 when this is to be divided into groups. The keys can be divided into a maximum of 9 groups each containing 1-16 keys. The function of the element with one key in a group differs from its function with groups of 2 - 16 keys. The keys on the keyboard are numbered as shown in the figure “Numbering of keyboard keys”.

Group Function, Several Keys

The keys on the keyboard are divided into groups of keys which function with group tripping. If a key in a group is pressed, the lamp in the key lights up, and the previously active key goes out.

The key to be included in the groups are specified at the MASK inputs. Each key has a corresponding bit, bit 0 corresponding to key 1, bit 1 corresponding to key 2 and so on, see example in Figure 129. The inputs EN are used to block group tripping. If EN = 0, no keys within the group are scanned and earlier values will remain at the output POSG.

The R input clears the group value and extinguish active lamps. This input is used where several keyboards are connected in a common group. R is not sensed when a key within the group is depressed. This is to prevent the clearing of POSG by the ACTC of the group with feedback as described below.

The outputs POSG specify which key in the group was pressed last, that is, the key with an active lamp in the group. POSG can adopt the value 0 for a reset group and 1 - (number of keys in the group). POSG=1 if the first key in the group is the last key pressed, and so on. If for example keys 1, 5, 9 and 13 are included in the group, POSG=1 corresponds to key 1, POSG=2 corresponds to key 5 and etc. The input signals R clear POSG in the relevant group.

After power failure, none of the keys will light up and the value of POSG will be 0 until one of the keys is pressed. When the ACT outputs give a pulse during one execution each time a key in the corresponding group is depressed. This output can be used if several keyboards are grouped together. The output signals ACTC from the different groups on the different keyboards are gated together with an OR element, the output of which is connected to R on corresponding groups.

Example of MASK calculation

Group	Keys
Ex.Groups 1	Keys 1, 2, 5, 6
Groups 2	Keys 3, 4, 7, 8
Groups 3	Key 16
→Groups 1	0 0 1 1 0 0 1 1→51=MASK 1
Groups 2	11 0 0 1 1 0 0→204=MASK 2
Groups 3	1 and 15 zeroes→32768=MASK 3

Group Function, One Key

A keyboard can contain keys which are controlled individually and are not included in any group. In these cases, the functions of the signals MASK, EN, POSG, ACT, R and ACTC are different from those described above. MASK indicates the key which is controlled. If only one bit is set in MASK, a single key function is obtained automatically. EN is used to control the lamp in the key specified. The lamp lights up if EN is set to 1. The output POSG is set to 1 if the key is pressed; otherwise it is 0.
The output ACT is normally used to sense the state of the key. The output ACT is set if the key is pressed; otherwise it is reset. With single key function, the signal R is not scanned. With single key function, ACTC is 0.

Error Supervision

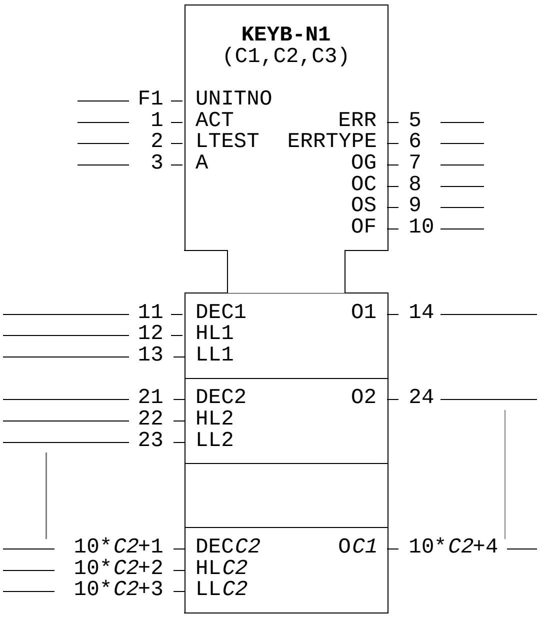
When the element indicates an error, the error output ERR is set, and the type of error is indicated at the same time at the ERRTYPE output. The element can indicate the following error types:

0=	The element has not detected an error (ERR=0)
1=	Controller board to specified MasterView 100 bus missing
2=	Controller board to specified MasterView 100 bus faulty
3=	Addressed MasterView 100 unit is missing or faulty
15=	Illegal value at function parameter UNITNO.

Numerical Keyboard KEYB-N1

Summary

KEYB-N1 (**KEYB**oard-**N**umeric **1**) is used for numerical entry from a numeric keyboard DSHM 120.



PC element KEYB-N1

Call DISP (C1, C2, C3)

Parameter	Significance	Permissible values
C1	Controls the usage of function keys F4 and F5. C1=1 means that F4/F5 are used for increase/ decrease. C1=0 means that F4/F5 are used as ordinary function keys.	0 - 1
C2	Number of values that can be handled by the element	1 - 9
C3	Data type of the values to be handled	B, I, IL, R, T, TR

Terminal Description

No	Name	Type	Description
F1	UNITNO	FI	**UNIT N**umber. Address for MasterView 100 unit. The hundreds and thousands indicate the number of the MasterView 100 bus, the units and tens indicate the number of the node.
1	ACT	IB	**ACT**ivate. Input for activation of display and scanning of values.
2	LTEST	IB	**L**amp **TEST.** Input for activation of lamp test.
3	A	II	**A**ddress. Address input.
5	ERR	OB	**ERR**or. Output for indication of error.
6	ERRTYPE	OI	**ERR**or **TYPE**. Output to determine type of error.

Terminal Description

No	Name	Type	Description
7	OG	OI	**O**utput **GR**oup. Output for the number of the last function key pressed.
8	OC	OB	**O**utput **C**lear. Set to 1 when CLEAR is pressed if element is activated.
9	OS	OB	**O**utput **S**end. Set to 1 when SEND is pressed if element is activated.
10	OF	OB	**O**utput **F**unction. Set to 1 when any function key is pressed.
11	DEC1	II	**DEC**imals **1**. Input for the number of decimal places to be displayed for value 1.
12	HL1	I*C3*	**H**igh **L**imit 1. Input for upper limit for value 1.
13	LL1	I*C3*	**L**ow **L**imit **1**. Input for lower limit for value 1.
14	O1	O*C3*	**O**utput **1**. Output for value 1.
21	DE*C2*	II	**DEC**imals **2**. Input for the number of decimal places to be displayed for value 2.
22	HL2	I*C3*	**H**igh **L**imit **2**. Input for upper limit for value 2.
23	LL2	I*C3*	**L**ow **L**imit **2**. Input for lower limit for value 2.
24 :	O2	O*C3*	**O**utput **2**. Output for value 2.
10×*C2*+1	DEC*C2*	II	**DEC**imals ***C2***. Input for the number of decimals to be presented for value *C2.*
10×*C2*+2	HL*C2*	I*C3*	**H**igh **L**imit. ***C2***. Input for upper limit for value *C2.*
10×*C2*+3	LL*C2*	I*C3*	**L**ow **L**imit ***C2***. Input for lower limit for value *C2.*
10×*C2*+4	O*C2*	O*C3*	**O**utput ***C2***. Output for value *C2.*

Function

KEYB-N1 is used for entry of numerical values from numeric keyboard DSHM 120. The element handles entry from keys, updating of the numerical display and control and supervision of function keys. Data can be entered by giving a completely new value or by increasing or decreasing the previous value. The element is all the time supervising that the selected value at output Ox is within the limits given by inputs LLx and HLx. Several KEYB-N1 can work with the same panel units at different times, e.g. when different data types are to be handled or when more than nine values are required. The element also handles the five function keys on the keyboard. The handling is done using group release. Since keyboards with membrane keys give a poor ’feel’ for whether and when a key is pressed, on detecting that a key has been unpressed, KEYB-N1 transmits a signal to a relay in the keyboard which gives a click sound as a positive indication that the key has been pressed. The display on DSHM 120 can also be controlled with the use of elements DISP and DISP-SEG.

Activation, Lamp Test and Addressing

Three inputs, ACT, LTEST and A are provided for control of the element function. If ACT is set to 1, the element presents values on the numerical display. All of the keys are monitored and the lamps in the keys are updated. If ACT=0, there is no presentation on the display and this can therefore be updated by some other element. The function keys are supervised and OG is updated. The lamps in the keys are not updated. A has no effect on the function of the element if ACT is set to 0. If LTEST is 1, all the segments in the numerical display and all the lamps in the keyboard keys are lit to test their function. Input A gives the address of the value to be processed. Values outside the interval 1 - C2 result in software error. (ERR is set to 1.) Addresses lower than 1 or greater than C2 are not treated.

Entry and Display of Values

In the case of values of data type R, T and TR, the value is displayed on the numerical display with the number of decimal places specified by the input DEC. DEC is ignored for other data types. If the increase/decrease key is pressed, the value is increased/ decreased while the key is held pressed as follows:

On decrease: The lowest decade is reduced to 0, after which the next decade is decreased to 0 etc.

On increase: The lowest decade is increased to 10 (9+memory digit), after which the next decade is increased to 10 (9+memory digit) etc.

If the increase/decrease key is released and pressed again, the increase/decrease begins from the lowest decade again. The value can only be changed to the limits specified by the inputs LL and HL. A completely new value can be entered in an entry phase. An entry phase begins when one of the keys 0-9, ”–”, CLR or ”.” is pressed. The entry phase is concluded by pressing SEND or CLR+CLR. The entry phase is indicated by the lamp in the SEND key being lit. The output O is not updated until SEND is pressed. The ”.” key is scanned only for values of type R, T and TR. Values of type R and TR with more decimal places than specified at the DEC input cannot be entered. When entering and presenting time, DEC specifies the number of decimal places for parts of second. If DEC=0, the most significant figure indicates hours

E.G.: DEC=0

Value entered	Interpreted as	Shows as
12.34.56	12.34.56.0000	12.34.56
12.34	12.34.00.0000	12.34.00
12.	12.00.00.0000	12.00.00
.12	00.12.00.0000	00.12.00

E.G.: DEC=3

Value entered	Interpreted as	Shows as
12.34.56	00.12.34.5600	12.34.560
12.34	00.12.34.0000	12.34.000
12.	00.12.00.0000	12.00.000
.12	00.00.12.0000	00.12.000

If CLR is pressed during the entry phase, the numerical display is cleared and the value entered is set to 0. New values can be given. If CLR is pressed again, with no intermediate key press, the entry phase is aborted without affecting the value of output O. SEND is pressed to conclude the entry phase and update output O with the entered value. KEYB-N1 checks that the value entered is within the specified limits. If any limit is exceeded, the numerical display flashed with incorrect value, and the output is not updated. With increase/decrease, the output is updated continuously, and the limits cannot be exceeded. The entry phase is aborted if the value at the address input A is changed.

Function Keys

There are five function keys on the keyboard used with KEYB-N1. These are in a column at the extreme left on the DSHM 120, see the figure below. If increase/decrease keys are to be used, that is, if call parameter *C1* is 1, the number of function keys on the keyboard is reduced to three.

The function keys are handled with group tripping, that is, the LED lights up in the last key pressed. The value at output OG indicates which key was last pressed with a value in the range 1 - (number of keys in the group). After power-on and before any function key has been pressed, the value at OG is 0 and no LED is lit.

F1	SEND		CLR
F2	1	2	3
F3	4	5	6
INCR/F4	7	8	9
DECR/F5	-	0	•

Keyboard layout, DSHM 120.

Initialization

At power-on, all digit positions on the display are blanked by all the elements it works with, to ensure that positions not controlled by any element are blanked.

Error Supervision

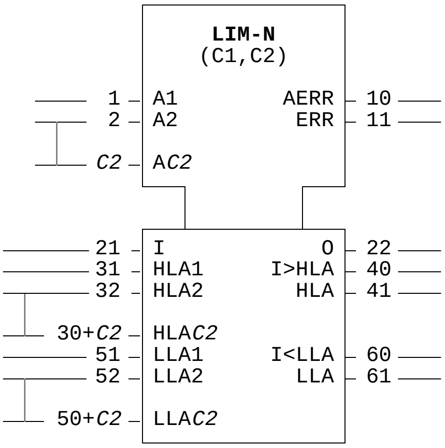
When the element indicates an error, the error output ERR is set, and the type of error is indicated at the same time at the ERRTYPE output. The element can indicate the following error types:

0=	The element has not detected an error (ERR=0)
1=	Controller board to specified MasterView 100 bus missing
2=	Controller board to specified MasterView 100 bus faulty
3=	Addressed MasterView 100 unit is missing or faulty
13=	Illegal value at input DECx
14=	Illegal values at input A
15=	Illegal values at function parameter UNITNO
16=	Value at terminal LL is higher than value at terminal HL

Limiter LIM-N

Summary

LIM-N (**LIM**iter **-** 1-of-**N** address) is used for limitation of integers, real numbers or time values. Several limit values can be selected.



PC element LIM-N

Call LIM-N (C1, C2)

Parameter	Significance	Permissible values
C1	Data type	I, IL, R, T, TR
C2	Number of optional limit values	1 - 9

Terminal Description

No	Name	Type	Description
1	A1	IB	**A**ddress **1**. Input which, when set, limits the output O to the limit values connected to the inputs HLA1 and HLL1.
2	A2	IM	**A**ddress **2**. Input which, when set, limits the output O to the limit values connected to the inputs HLA2 and HLL2.
: *C2*	A*C2*	IB	**A**ddress ***C2***. Input which, when set, limits the output O to the limit values connected to the inputs HLA*C2* and LLA*C2*.
10	AERR	OB	**A**ddress **ERR**or. Output which is set when 2 or more of the inputs A1 - A*C2* are set.
11	ERR	OB	**ERR**or. Output which is set when the limit for high level is less than the limit for low level.
21	I	I*C1*	**I**nput. Input to which the signal to be limited is connected.
22	O	O*C1*	**O**utput. Output for the limited signal.
31	HLA1	I*C1*	**H**igh **L**imit **A**ddress **1**. Input for upper limit value which limits the signal when the input A1 is set.

Terminal Description

No	Name	Type	Description
32	HLA2	I*C1*	**H**igh **L**imit **A**ddress **2**. Input for upper limit value which limits the signal when the input A2 is set.
: 30+*C2*	HLA*C2*	IC1	**H**igh **L**imit **A**ddress ***C2***. Input for upper limit value which limits the signal when the input AC2 is set.
40	I>HLA	OB	**I**nput>**H**igh **L**imit **A**ddress. Output which is set when the upper limit of the element is reached.
41	HLA	OC1	**H**igh **L**imit **A**ddress. Output which specifies the upper limit where limiting begins.
51	LLA1	I*C1*	**L**ow **L**imit **A**ddress **1**. Input for lower limit value which limits the signal when the input A1 is set.
52	LLA2	I*C1*	**L**ow **L**imit **A**ddress **2**. Input for lower limit value which limits the signal when the input A2 is set.
: 50+*C2*	LLA*C2*	IC1	**L**ow **L**imit **A**ddress ***C2***. Input for lower limit value which limits the signal when the input A*C2* is set.
60	I<LLA	OB	**I**nput<**L**ow **L**imit **A**ddress. Output which is set when the lower limit of the element is reached.
61	LLA	O*C1*	**H**igh **L**imit **A**ddress. Output which specifies the lower limit where limiting begins.

Function

LIM-N is used to limit up to 9 different limits. Boolean output signals are given when the output is limited.

Selection of Limit Value

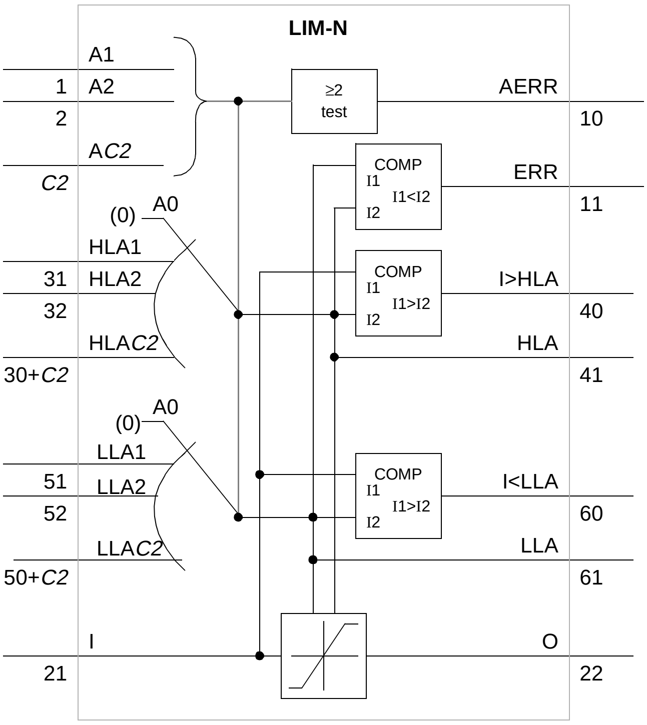
Which of the limit value inputs HLA1 - HLA*C2* or LLA1 - LLA*C2* is to limit the value at output O is selected with the inputs A1 - A*C2*. If the input A1 is 1, the output is limited by HLA1 and LLA1, if A2 is 1, the output is limited by HLA2 and LLA2, etc. If none of the inputs A1 - A*C2* is 1, O is limited to the data value 0. If 2 or more of the inputs A1 - A*C2* are set at the same time, the output is limited by the limit values corresponding to the lowest numbered set input. The error signal output AERR is set at the same time.

Limiting

When the input I exceeds the selected limit, the output O is limited to the limit value. One of the outputs I>HLA or I<LLA will then be set depending on which limit was exceeded. The value of the current limits for high and low level where limiting begins can be read at the outputs HLA and LLA.

Supervision of Limit Values

The element checks that the limit value HLA is greater than the limit LLA. If HLA is less than LLA, the error signal output ERR is set. The output O and the limit value outputs I>HLA and I<HLL are retained from the sample before that in which the error status developed.

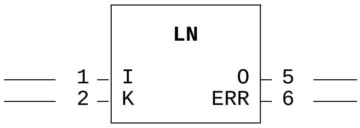


Function diagram

Natural Logarithm LN

Summary

LN (**L**ogarithm **N**atural) is used to calculate the natural logarithm of a real number. The result can be multiplied by an optional value.



PC element LN

Call LN

Terminal Description

No	Name	Type	Description
1	I	IR	**I**nput. Input for a number of which the logarithm is to be calculated.
2	K	IR	Input for multiplication factor K.
5	O	OR	**O**utput. Output for the product of the value at input K and the logarithm of the value at input I.
6	ERR	OB	**ERR**or. Output which is set if the value at input I is less than or equal to 0.

Function

The natural logarithm of the value at input I is calculated. The result is multiplied by the value at input K. The product is stored at the output O.

Supervision

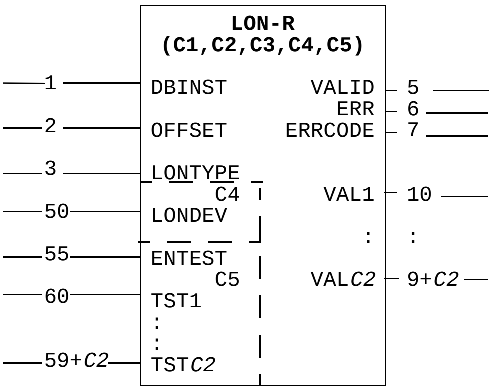
If the value at the input I is negative or 0, the error output ERR is set to 1.

Receive data from LONWORKS Network LON-R

Summary

LON-R is used for receiving data which originates from an input Network Variable, defined by a LONNVI or LONMNVI data base element.

A data simulation mode is available through call parameter C5.



Call LON-R(C1,C2,C3,C4,C5)

Call Parameters

Parameter	Significance	Permissible values	Default value
C1	Output VALue terminals data type	A, B, I, IL, R, T	-
C2	Number of data VALues	1.......32	1
C3	If C1=A: Number of bytes in Array	0......40	0
C4	C4=1 if LONMNVI is used (for INSUM)	0, 1	0
C5	C5=1 if test terminals are to be used	0, 1	0

Terminal Description

Number	Name	Type	Description
1	DBINST	II	**D**ata **B**ase **INST**ance; A parameter defining the instance number of related data base element (LONNVI or LONMNVI), alternatively, a symbolic connection to the element. (For example =LONNV147)
2	OFFSET	II	**OFFSET** in bytes for the start of the wanted (first) data value in this NV
3	LONTYPE	IA2	**LON TYPE** according to conversion table below. This terminal decides the data conversion type. You enter the value for LONTYPE as a two character string for example D=’FP’.
5	VALID	OB	Data **VALID**ation; 1 = data is valid. The VALID terminal is set TRUE as soon as a new reading is successfully made. It’s set FALSE when the NV status information says so.

Number	Name	Type	Description
6	ERR	OB	**ERR**or. This output terminal is set if an error is found.
7	ERRCODE	OI	**ERR**or **CODE**. See section "Error codes".
10	VAL1	*C1*	First received data **VAL**ue.
:	:	:	
9 + *C2*	VAL*C2*	*C1*	Last received data **VAL**ue
50	LONDEV	II	**LON DEV**ice. A parameter defining the instance number of related data base element (LONDEV), alternatively, a symbolic connection to the element. (for example =LONDEV26).This terminal defines the sending device when LONMNVI is used as network variable. Only if C4=1
55	ENTEST	IB	**EN**able **TEST**ing Only if C5=1
60	TST1	*C1*	First **T**e**ST** input Only if C5=1
:	:	:	
59+*C2*	TST*C2*	*C1*	Last **T**e**ST** input Only if C5=1

Function

LON-R receives data from the Network variable defined by the data base element LONNVI or LONMNVI pointed out by the terminal DBINST.

In case of Network Variables bounded “many to one” the LONMNVI is used. It is thus necessary to define the sending device. This is done with LONDEV terminal and call parameter C4.

The terminal OFFSET gives the address adjustment (in bytes) from the start of the network variable to where the data component starts.

Terminal LONTYPE gives together with call parameter C1 the wanted conversion procedure (one line in the conversion table) which settles both how many bytes are affected and the transformation procedure. If different data types are possible in the application software then the conversion table gives alternative values of PC element call parameter C1.

If data doesn’t fit the corresponding data type size the least significant bits are always used first when reducing the size and zeroes are added in the most significant positions when filling up. The value for OFFSET and LONTYPE are found in the SNVT table below.

The VALID terminal of the LON-R element is set if the read data is fresh. This is given by the Status bytes of this Network Variable which gets its information from the LONWORKS device when polling one of its NVs. The ERR and ERRCODE output terminals of LON-S are set in the same way.

It is possible to use several LON-R to read data from one network variable.

Error Codes

The following error codes may appear on the ERRCODE terminal:

0. No error.

1. Data base element given by DBINST or LONDEV is not found.
2. Data base element marked ’Not active’.
3. Invalid connection to DB element.
4. Data base element of incorrect type.
5. Data base element error marked.
6. Data base element not implemented.
7. Data base element not in service.
9. Data base element is not configurated correctly.
11. Offset plus data length exceeds the Network Variable length value.
12. No bytes in array.
13. Bit field conversion conditions not met.
14. Communication area occupied by another PC element.
15. Three consecutive reading of hardware failures, - output is ‘frozen’.
16. No NV status information available, but values are updated
17. Bus error. CI572/CI573 not accessible.
18. LONDEV address not found in LONMREF table.
21. NV Status: Passive
23. NV Status: Error
24. NV Status: Unconfigured
30. NV Status Error: No parameter configured
31. NV Status Error: Parameter not correct
32. NV Status Error: Data not accessible
33. NV Status Error: Invalid data
35. NV Status Error: Heart-beat time-out
36. NV Status Error: Run-time error
37. NV Status Error: Update failed. (Check binding of the Network Variable).
56. Unknown kind of LONTYPE.
57. Illegal data conversion is defined
58. Data conversion results in Overflow. Some data is lost, but values are updated.

111-119. Errors 1-9 referring to DB element LONNVI.
121-129. Errors 1-9 referring to DB element LONMNVI.
131-139, 159. Errors 1-9 referring to DB element LONMREF.
141-149. Errors 1-9 referring to DB element LONDEV

For all error codes except for 16 and 58, ERR terminal is set and value output terminals are cleared.

Conversion of data between LONWORKS and AMPL data types

Only some of the LONWORKS data types defined in the SNVT list, fit directly into ABB Master data types. The other must be represented in AMPL, ABB Master Programming Language with data types containing normally more bytes than they have in SNVT list. The table below shows the data types used.

The conversion transformation procedure is unambiguously decided by the PC element terminal LONTYPE in combination with the Call parameter C1 and C3 giving the AMPL data type; thus specifying one line in the table below.

Conversion table for terminal LONTYPE and call parameter C1.

Terminal LONTYPE (1)	LONWORKS Range	LON bytes	C1 AMPL types (2)
US, Unsigned short	0.......255	1	I, IL
US, Unsigned short	Eight packed Booleans	1	B
FT, State, (False/True)	False (0x00), True (0xFF)	1	B, I
SS, Signed short	-128........127	1	I, IL, R
UL, Unsigned long	0.......65 535	2	IL, R
UL, Unsigned long	Sixteen packed Booleans	2	B
SL, Signed long	-32 768.....32 767	2	I, IL,R
FP, Floating Point	$- 1 \bullet 10^{38}$ .... $+ 1 \bullet 10^{38}$	4	R, IL
S3, Signed 32 bits	-2 147 483 648...........2 147 483 647	4	IL, R
SR, Structure	Time: Hour: minute: second, millisecond	5	T
SR, Structure	String of bytes (Array)	C3	A
<XY>, Bit field (A value of <X> bits	0≤<X>≤7; 0≤<Y>≤7; <X> + <Y>≤8; with an offset of <Y> bits within a byte)	1	I, B, IL
99 Only status reading - no conversion. Only ERR and ERRCODE outputs are updated.			

(1) The SNVT type two character abbreviations are intended for the PC element terminal LONTYPE. Please observe that a long data type consists of two bytes in LONWORKS but four in AMPL. <X> and <Y> are both digits between 0 and 7 with a sum of maximum 8. For Booleans: C2 = <X>;

(2) The first stated type in the table column is the recommended one.

For detailed information on existing Standard Network Variable Types and ABB INSUM specific Network Variables see SNVT-Table below. In it all LONWORK types and offsets, range factors and lengths are stated for all SNVT and INSUM data values.

For further information see documentation from LONMARK, ECHELON.

SNVT range.....Range factor

Most Standard Network Variable Types, SNVTs also contain a "range factor" which is **not** transmitted on the LONWORKS network as this would result in nonexisting data types, for example SNVT_lev_percent has the range "Signed long/200" (that is "range factor" 200) which means that it has the range of -163.84.....+163.83 with a resolution of 0.005%.
This SNVT specific range factor is **not** transmitted. In this case the transmitted value (-32768 ... ...+32766) is 200 times bigger

than the value of the NV data value and **not** adjusted in the PC elements. **The range factor is left to the application engineer to take care of.** (163.83 → hexadecimal "7FFE", is the maximum as "7FFF" for this SNVT is defined invalid.)

If the LONWORKS data string doesn’t fit the corresponding data type the least significant bits are always used first when reducing the size and zeroes are added in the most significant positions when filling up.

Overflow conversion

When overflow occurs in data type conversion the value is set to its nearest limit value.

Packed Booleans Unsigned Short conversion

Terminal "VAL1" corresponds to the most significant bit in the first byte and so on.
Terminal "VAL8" corresponds to the least significant bit in the first byte.
Terminal "VAL9" corresponds to the most significant bit in the second byte and so on.

Packed Booleans Unsigned Long conversion

Terminal "VAL1" corresponds to the most significant bit in the first word and so on.
Terminal "VAL16" corresponds to the least significant bit in the first word.
Terminal "VAL17" corresponds to the most significant bit in the second word and so on.

Time conversion

The AMPL data type T defines time from 0 to 23:59:59.999 (format hours:minutes:seconds.millisecounds).

It will be converted into/from the SNVT format: Hours (one byte),
Minutes (one byte), Seconds (one byte) and Milliseconds (two bytes).

LONWORKS structure - Array

The AMPL array cells are packed/unpacked after each other,
one array after the other, in the SNVT structure.

Bit field conversion

Both <X> and <Y> are digits between 0 and 7.
<X> tells the number of bits concerned within the byte
and <Y> tells the start offset in bits from the most significant position in the byte.

Condition for all Bit field conversions: <X> + <Y> ≤ 8.

Additional condition for conversions to/from Booleans: C2 = <X>.

When converting to/from I and IL only the <X> least significant bits are used.

Booleans are all packed/unpacked into/from one and the same SNVT byte
while consecutive Integers and Integer Longs use consecutive SNVT bytes;
(bits from one Integer are put to/from one SNVT byte).

For example: A card reader conforming to ISO 7811 will read standard financial
transaction cards (credit cards and Automatic Teller Machine cards) where digits

are made up of four bits each and two of them are packed into one byte
according to the TYPE SNVT_magcard (#86).

The first digit will have type "40" and contain the most significant bits
and the second digit type "44" and contain the least significant bits.

The third digit will have SNVT type "40" and the 4th SNVT type "44" and so on.

Each digit can be transformed from/into Booleans or Integers or Integer Longs.

The Booleans "True","True","False" and "True" will with LONTYPE ’40’
produce the byte hexadecimal "D0", while LONTYPE ’44’ will give "0D".
If two LON-S PC-elements making these conversions on the same Network
variable, NV also have the same OFFSET they will operate on the same LONWORKS byte
which will then become "DD".

Integers with value "13" or hexadecimal "D" will produce the same.
Also Integers with for example the values "125" (hexadecimal "7D") and "3021" ("BCD")
will produce the same as only the 4 least significant Integer bits are used.

State

SNVT type State:"False" corresponds to the Boolean "False" and the Integer Ox00.
SNVT type State:"True" corresponds to the Boolean "True" and the Integer OxFF.

Floating point conversion

The floating point format is according to ANSI/IEEE standard 754-1985. The range of AMPL data type R is however limited to $\pm 9.2 \times 10^{18}$. Values outside this range are limited.

Official Standard Network Variable Types (SNVT) list:

SNVT nr	Name	Conversions defined by LONTYPE	Range	Offset	R.f.	Length	Notes
114	SNVT_address	UL	0x4000...0xF1FF	0	1	2	Neuron chip address
88	SNVT_alarm	SR (6)	6 bytes	0		29	Alarm, Node location
		UL	0....65535	6	1		Object ID within node
		US	0.....12	8	1		Alarm type
		US	0.......3	9	1		Priority level
		UL	1....#SNVTs	10	1		SNVT ID
		4xUS, (2xUL)		12			Value specific to SNVT
		UL	0.....3000	16	1		Year. (0=not specified)
		US	0....12	18	1		Month. (0=not specified)
		US	0.....31	19	1		Day. (0=not specified)
		SR (5)	5 bytes	20	1		Time:hour,minute,second,millisecond
		(US)	0.....23	20	1		Hour
		(US)	0.....59	21	1		Minute
		(US)	0.....59	22	1		Second
		(UL)	0.....999	23	1		Milliseconds
		4xUS, (2xUL)		25			Alarm limit specific to SNVT
4	SNVT_angle_vel	SL	-3276.8....3276.7	0	10	2	Angular velocity (radians/sec)
50	SNVT_angle_vel_f	FP	-1E38...1E38	0	1	4	Angular velocity (radians/sec.)
102	SNVT_rpm	UL	0....65534	0	1	2	Velocity (rpm) 0xFFFF=invalid data
110	SNVT_area	UL	0.....13.1068	0	5000	2	m^2 0xFFFF=invalid data
7	SNVT_char_ascii	US	0....255	0	1	1	Ascii character
36	SNVT_str_asc	SR (31)	0...30 characters	0		31	Array of 31 bytes
37	SNVT_str_int	SR (31)		0		31	Intended for wide characters à two bytes.
127	SNVT_chlr_status	US		0	Enum		Chiller run mode
		US		1	Enum		Chiller operation mode
		US	6 bits	2	1		Chiller state
70	SNVT_color	UL		0		6	L_star
		SL		2			a_star
		SL		4			b_star

SNVT nr	Name	Conversions defined by LONTYPE	Range	Offset	R.f.	Length	Notes
29	SNVT_ppm	UL	0....65535	0	1	2	ppm, parts per million
58	SNVT_ppm_f	FP	0....1E38	0	1	4	Floating ppm, parts per million
8	SNVT_count	UL	0....65535	0	1	2	Counts
51	SNVT_count_f	FP	0....1E38	0	1	4	Floating counts
9	SNVT_count_inc	SL	-32768....32767	0	1	2	Signed counts
52	SNVT_count_inc_f	FP	-1E38....1E38	0	1	4	Signed floating counts
89	SNVT_currency	US	1....54	0	Enum	6	Currency type
		SS	-128....127	1	1		Power_of_10
		S3	-1E38....1E38	2	1		Credit positive, Debet negative
1	SNVT_amp	SL	-3276.8....3276.7	0	10	2	Current ampère
139	SNVT_amp_ac	UL	0...65534 A	0	1	2	0xFFFF = Unknown value
48	SNVT_amp_f	FP	-1E38....1E38	0	1	4	Floating current ampère
2	SNVT_amp_mil	SL	-3276.8....3276.7	0	10	2	Current milliampère
10	SNVT_date_cal	-	-	-	-		Use SNVT_timestamp instead
11	SNVT_date_day	US	0....6	0	Enum	1	0=Sunday....6=Saturday
120	SNVT_defr_mode	US	0....2	0	Enum	1	Type of defrost
121	SNVT_defr_term	US	0....3	0	Enum	1	Defrost termination
122	SNVT_defr_state	US	0....4	0	Enum	1	Defrost progress
100	SNVT_density	UL	0....32767.5	0	2	2	kg/m^3
101	SNVT_density_f	FP	0.....1E38	0	1	4	Floating kg/m3
103	SNVT_hvac_emerg	US	0....4	0	Enum	1	Emergency mode
135	SNVT_arth_pos	10	0..1	0	1 bit	10	Latitude direction
		11	0..1	0	1 bit		Longitude direction
		62	0..90	0	1		Latitude degrees. 0xFFFF=undefined
		UL	00.000.....59.999	1	1000		Latitude minutes 0xFFFF=undef.
		US	0....180	3	1		Longitude degrees 0xFFFF=undf
		UL	00.000.....59.999	4	1		Longitude minutes 0xFFFF=undef.
		FP	-1E30....1E30	6	1		Height above sea (meter). -1E38=undefined
13	SNVT_elec_kwh	UL	0....65535	0	1	2	kWh
14	SNVT_elec_whr	UL	0....6553.5	0	10	2	Watt-hours
68	SNVT_elec_whr_f	FP	0....1E38	0	1	4	Floating watt-hours

SNVT nr	Name	Conversions defined by LONTYPE	Range	Offset	R.f.	Length	Notes
67	SNVT_btu_f	FP	0....1E38	0	1	4	Floating thermal energy btu
5	SNVT_btu_kilo	UL	0....65535	0	1	2	Thermal energy kilo btu
6	SNVT_btu_mega	UL	0....65535	0	1	2	Thermal energy mega btu
118	SNVT_evap_state	US	0....2	0	Enum	1	Evaporation state
90	SNVT_file_pos	4xUS	0....232-1	0	1	6	Four bytes address pointer
		UL	0....65535	4	1		Transfer length in bytes
73	SNVT_file_req	US	0....6	0	Enum	12	File request order
		UL	0....65535	1	1		File index in directory
		UL	0....65535	3	1		Receive time-out in milliseconds
	Alternative #1	US		5	1		Type
		10	1 bit	6	1		Domain
		71	7 bits	6	1		Node
		40	4 bits	7	1		
		44	4 bits	7	1		Retry
		40	4 bits	8	1		
		44	4 bits	8	1		Tx-timer
		US		9	1		Subnet
	Alternative #2	10	1 bit	5	1		Type
		71	7 bits	5	1		Size
		10	1 bit	6	1		Domain
		71	7 bits	6	1		
		40	4 bits	7	1		
		44	4 bits	7	1		Retry
		40	4 bits	8	1		
		44	4 bits	8	1		Tx-timer
		US		9	1		Group
		SS	0....1	10	1		Authentication protocol on
		SS	0....1	11	1		Priority messaging on

SNVT nr	Name	Conversions defined by LONTYPE	Range	Offset	R.f.	Length	Notes
74	SNVT_file_status	US	0....11	0	Enum	27	Status code
		UL	0....65535	1	1		Directory size
		UL	0....65535	3	1		Index of selected file
	Alternative #1	A (16)		5			Array of 16 bytes of 1 character each
		4xUS		21			Size
		UL		25			Type
	Alternative #2	6xUS		5		(14)	Domain identity
		US		11			Domain length
		US		12			Subnet
		US		13			Node
133	SNVT_fire_indcte	US		0	Enum	1	Fire indicator
132	SNVT_fire_init	US		0	Enum	1	Fire initiator
130	SNVT_fire_test	UL	0....65534	0	1	3	Object identity
		US		2			Fire test
15	SNVT_flow	UL	0....65534	0	1	2	Liters/second 0x7FFF=invalid data
53	SNVT_flow_f	FP	-1E38....1E38	0	1	4	Floating flow
16	SNVT_flow_mil	UL	0....65535	0	1	2	milliliters/second
75	SNVT_freq_f	FP	0....1E38	0	1	4	Floating frequency Hertz
76	SNVT_freq_hz	UL	0....6553.5	0	10	2	Frequency Hertz
77	SNVT_freq_kilohz	UL	0....6553.5	0	10	2	Frequency kiloHertz
78	SNVT_freq_milhz	UL	0....6.5535	0	10000	2	Frequency milliHertz
91	SNVT_muldiv	UL	0....65535	0	1	4	Gain factor: multiplier
		UL	0....65535	2	1		Gain factor: divisor
71	SNVT_grammage	UL	0....6553.5	0	10	2	gramme
72	SNVT_grammage_f	FP	0....1E38	0	1	4	Floating gramme
108	SNVT_hvac_mode	US	0....9	0	Enum	1	HVAC,Heating,Ventilation,Air Conditioning
111	SNVT_hvac_overid	US	0....7	0	Enum	5	HVAC State
		SL	-163.84....163.83	1	200		HVAC percent
		UL	0....65534	3	1		HVAC flow

SNVT nr	Name	Conversions defined by LONTYPE	Range	Offset	R.f.	Length	Notes
112	SNVT_hvac_status	US	0....9	0	Enum	12	HVAC mode
		SL	-163.84....163.83	1	200		HVAC Primary heat output
		SL	-163.84....163.83	3	200		HVAC Secondary heat output
		SL	-163.84....163.83	5	200		HVAC Cooling output
		SL	-163.84....163.83	7	200		HVAC Economizer output
		SL	-163.84....163.83	9	200		HVAC Fan output
		US	0....1	11	0		HVAC alarm
145	SNVT_hvac_type	US	0....9	0	Enum	1	
79	SNVT_lux	UL	0....65535	0	1	2	Lux Illumination
69	SNVT_config_src	US	0....1	0	1	1	Configuration source
17	SNVT_length	UL	0....6553.5	0	10	2	Length meter
54	SNVT_length_f	FP	0....1E38	0	1	4	Floating length meter
18	SNVT_length_kilo	UL	0....6553.5	0	10	2	Length kilometer
19	SNVT_length_micr	UL	0....6553.5	0	10	2	Length micrometer
20	SNVT_length_mil	UL	0....6553.5	0	10	2	Length millimeter
21	SNVT_lev_cont	US	0....100	0	2	1	Percent
55	SNVT_lev_cont_f	FP	0....100	0	1	4	Floating percent
22	SNVT_lev_disc	US	0....4	0	Enum	1	Discrete levels
81	SNVT_lev_percent	SL	-163.84....163.83	0	200	2	Level percent. 0x7FFFF=invalid data
86	SNVT_magcard	40	4 bits	0		20	Two digits per byte
		44	4 bits	0			
		40	4 bits	19			Totally 40 digits in 20 bytes
		44	4 bits	19			
80	SNVT_ISO_7811						Use SNVT_magcard instead
23	SNVT_mass	UL	0....6553.5	0	10	2	grams
56	SNVT_mass_f	FP	0....1E38	0	1	4	Floating grams
24	SNVT_mass_kilo	UL	0....6553.5	0	10	2	Kilos
25	SNVT_mass_mega	UL	0....6553.5	0	10	2	Metric tons
26	SNVT_mass_mil	UL	0....6553.5	0	10	2	Milligrams
82	SNVT_multiplier	UL	0....32.7675	0	2000	2	
92	SNVT_obj_request	UL	0....65535	0	1	3	ID of object within node
		US	0....15	2	Enum		Object request field

SNVT nr	Name	Conversions defined by LONTYPE	Range	Offset	R.f.	Length	Notes
93	SNVT_obj_status	UL	0....65535	0	1	6	ID of object within node
		4xUS	32 bits	2.....5			Packed Booleans
109	SNVT_occupancy	US	0....3	0	Enum	1	Area occupancy status
97	SNVT_override	US	0....2	0	Enum	1	
3	SNVT_angle	UL	0....65.535	0	1000	2	Angle radians
104	SNVT_angle_deg	SL	-359.98....360.00	0	50	2	Angle degrees 0x7FFF=invalid data
49	SNVT_angle_f	FP	-1E38....1E38	0	1	4	Angle floating radians
38	SNVT_telcom	US	0....20	0	Enum	1	Phone state
125	SNVT_ph	SL	-32.768...32.767	0	1000	2	PH acidity
126	SNVT_ph_f	FP	-1E38....1E38	0	1	4	PH acidity floating point
27	SNVT_power	UL	0....6553.5	0	10	2	Power watts
57	SNVT_power_f	FP	-1E38....1E38	0	1	4	Power floating watts
28	SNVT_power_kilo	UL	0....6553.5	0	10	2	Power kilowatts
98	SNVT_pwr_fact	SL	-1.0....1.0	0	20000	2	Power factor
99	SNVT_pwr_fact_f	FP	-1.0....1.0	0	1	4	Floating power factor
94	SNVT_preset	US	0...3	0	Enum	14	Use SNVT_scene(_cfg) instead
		UL	1....65534	1	1		Selector
		(S3)	0....4294967294	3	1		Value
		Warning: The signal above is a 32 bits positive number of four bytes.					
		UL		7	1		Day
		US		9	1		Hour
		US		10	1		Minute
		US		11	1		Second
		UL		12	1		Millisecond
30	SNVT_press	SL	-3276.8....3276.7	0	10	2	Pressure-gauge kilopascals
59	SNVT_press_f	FP	0....1E38	0	1	4	Pressure Pascals absolute
113	SNVT_press_p	SL	-32768....32767	0	1	2	Pressure pascals 0x7FFF=invalid data
136	SNVT_reg_val	S3	±2147483648	0	1	6	
		US	0....43	4	Enum		0xFF = Invalid value
		US	0.....7	5	1		Number of decimals

SNVT nr	Name	Conversions defined by LONTYPE	Range	Offset	R.f.	Length	Notes
137	SNVT_reg_val_ts	S3	±2147483648	0	1	13	
		US	0....43	4	Enum		0xFF = Invalid value
		30	0.....7	5	1		Number of decimals
		43	0....15	5	1		Status
		17	0.....1	5	1		Register state
		UL		6	1		Year
		US		8	1		Month
		US		9	1		Day
		US		10	1		Hour
		US		11	1		Minute
		US		12	1		Second
31	SNVT_res	UL	0....6553.5	0	10	2	Resistance ohms Ω
60	SNVT_res_f	FP	-1E38....1E38	0	1	4	Resistance ohms Ω
32	SNVT_res_kilo	SL	0....6553.5	0	10	2	Resistance kilo-ohms kΩ
117	SNVT_setting	US	0....5	0	Enum	4	Setting function field
		US	0....100%	1	2		Setting percentage of full scale
		SL	-359.98....360.00	2	50		Rotation angle degrees
115	SNVT_scene	US	0....1	0	Enum	2	Learn/recall scene
		US	1....255	1	1		Scene number
116	SNVT_scene_cfg	US	0....4	0	Enum	10	Function field
		US	1....255	1	1		Scene number
		US	0....100%	2	2		Setting percentage of full scale
		SL	-359.98....360.00	3	50		Rotation angle degrees
		UL	0.0....6553.4	5	10		Fade time seconds
		UL	0.0....6553.4	7	10		Delay time seconds
		US	0....255	9			Priority 0=highest
33	SNVT_sound_db	SL	-327.68....327.67	0	100	2	Sound level decibel
61	SNVT_sound_db_f	FP	-1E38....1E38	0	1	4	Floating sound level decibel
34	SNVT_speed	UL	0....6553.5	0	10	2	Speed meter/second
62	SNVT_speed_f	FP	-1E38....1E38	0	1	4	Floating speed m/s
35	SNVT_speed_mil	UL	0....65.535	0	1000	2	Speed m/s
83	SNVT_state	UL	16 bits	0	1	2	Sixteen Packed Booleans

SNVT nr	Name	Conversions defined by LONTYPE	Range	Offset	R.f.	Length	Notes
95	SNVT_switch	US		0		2	User defined value
		US		1			User defined state
39	SNVT_temp	UL	-274.0....6279.5	0	10	2	Tenth of Celsius degrees above -274°
105	SNVT_temp_p	SL	-273.17....327.66	0	100	2	Temperature Celsius. 0x7FFF=invalid
63	SNVT_temp_f	FP	-273.17....1E38	0	1	4	Floating temperature Celsius
131	SNVT_temp_ror	UL	0...32767	0	2	2	Temp. rate of rise. °C/min 0x7FFF=0
106	SNVT_temp_setp	SL	-273.17....327.66	0	100	12	Occupied cool 0x7FFF=invalid
		SL	-273.17....327.66	2	100		Standby_cool 0x7FFF=invalid
		SL	-273.17....327.66	4	100		Unoccupied_cool 0x7FFF=invalid
		SL	-273.17....327.66	6	100		Occupied_heat 0x7FFF=invalid
		SL	-273.17....327.66	8	100		Standby_heat 0x7FFF=invalid
		SL	-273.17....327.66	10	100		Unoccupied_heat 0x7FFF=invalid
119	SNVT_therm_mode	US	0....2	0	Enum	1	Temp control mode
12	SNVT_date_time						Use SNVT_time_stamp instead
64	SNVT_time_f	FP	0....1E38	0	1	4	Elapsed floating time
87	SNVT_elapsed_tm	UL	0....65534	0	1	7	Elapsed time: Day. 65535 means Null
		SR	5 bytes	2	1		Time: hour, minute, second, millisecond
		(US)	0....23	2	1		Elapsed time:Hour
		(US)	0....59	3	1		Elapsed time:Minute
		(US)	0....59	4	1		Elapsed time:Second
		(UL)	0....999	5	1		Elapsed time:Millisecond
124	SNVT_time_hour	UL	0...65535	0	1	2	Time hours
123	SNVT_time_min	UL	0...65534	0	1	2	Time minutes
107	SNVT_time_sec	UL	0.0....6553.4	0	10	2	Time seconds 0xFFFF=invalid data

SNVT nr	Name	Conversions defined by LONTYPE	Range	Offset	R.f.	Length	Notes
40	SNVT_time_passed						Use SNVT_elapsed_tm instead
84	SNVT_time_stamp	UL	0....3000 AD	0	1	7	Year. 0=not specified. 65535=Null date
		US	0....12	2	1		Month 0=not specified.
		US	0....31	3	1		Day 0=not specified.
		US	0....23	4	1		Hour
		US	0....59	5	1		Minute
		US	0....59	6	1		Second
134	SNVT_time_zone	S3	-86400....86400	0	1	15	Second offset from GMT. East = plus.
		US	0...2	4	Enum		Type of description of time zone
		US	0....23	5	1		Hour of start of Daylight Saving Time,DST
		US	0....59	6	1		Minute of start DST
		US	0....59	7	1		Second of start DST
	Alternative 1a	UL	0....365	8	1		Day of start DST
	Alternative 1b	UL	0....365	8	1		Day of start DST
	Alternative 1c	40	1....12	8	1		Month of start DST
	Alternative 1c	34	1.......5	8	1		Week of start DST
	Alternative 1c	US	0......6	9	1		Dateday of start DST
		US	0.....23	10	1		Hour of end of Daylight Saving Time,DST
		US	0....59	11	1		Minute of end DST
		US	0....59	12	1		Second of end DST
	Alternative 2a	UL	0....365	13	1		Day of end DST
	Alternative 2b	UL	0....365	13	1		Day of end DST
	Alternative 2c	40	1....12	13	1		Month of DST end
	Alternative 2c	34	1.......5	13	1		Week of DST end
	Alternative 2c	US	0......6	14	1		Dateday of DST end
128	SNVT_tod_event	US		0	Enum	4	Current state
		US		1	Enum		Next state
		UL	0....65535	2	1		Time in minutes to next states

SNVT nr	Name	Conversions defined by LONTYPE	Range	Offset	R.f.	Length	Notes
96	SNVT_trans_table	7xFP	-1E38....1E38	0	1	30	Seven translation values
		20	0....1	28	Enum		Interpolation type
		22	0....1	28	Enum		Interpolation type
		24	0....1	28	Enum		Interpolation type
		26	0....1	28	Enum		Interpolation type
		20	0....1	29	Enum		Interpolation type
		22	0....1	29	Enum		Interpolation type
		24	0....1	29	Enum		Interpolation type
143	SNVT_turbidity	UL	0.000....65.534	0	1000	2	0xFFFF = Unknown value
144	SNVTturbidity_f	FP	0..1E38	0	1	4	
41	SNVT_vol	UL	0....6553.5	0	10	2	Volume liters
65	SNVT_vol_f	FP	0....1E38	0	1	4	Floating volume liters
42	SNVT_vol_kilo	UL	0....6553.5	0	10	2	Volume kiloliters
43	SNVT_vol_mil	UL	0....6553.5	0	10	2	Volume milliliters
44	SNVT_volt	SL	-3276.8....3276.7	0	10	2	Voltage volts
138	SNVT_volt_ac	UL	0....65534 VAC	0	1	2	0xFFFF = Unknown value
45	SNVT_volt_dbmv	SL	-327.68....327.67	0	10	2	Voltage dB microvolts. µV
66	SNVT_volt_f	FP	-1E38....1E38	0	1	4	Floating volts
46	SNVT_volt_kilo	SL	-3276.8....3276.7	0	10	2	Kilovolts
47	SNVT_volt_mil	SL	-3276.8....3276.7	0	10	2	Millivolts
85	SNVT_zerospan	SL	-163.84....163.835	0	200	4	Zero-offset
		UL	0....32.7675	2	2000		Span multiplier
NV typenr	**Name**	**LON type**	**Range**	**Offset**	**R.f.**	**Length**	**Notes**
252	NV_current_report (NV_curr_report)	FP	-1E38.....1E38	0	1	22	Phase current ampére
		FP	-1E38.....1E38	4	1		Phase current ampére
		FP	-1E38.....1E38	8	1		Phase current ampére
		FP	-1E38.....1E38	12	1		Earth fault ampére
		UL	0.....65535	16	1		Current %
		UL	0.....65535	18	1		Current %
		UL	0.....65535	20	1		Current %

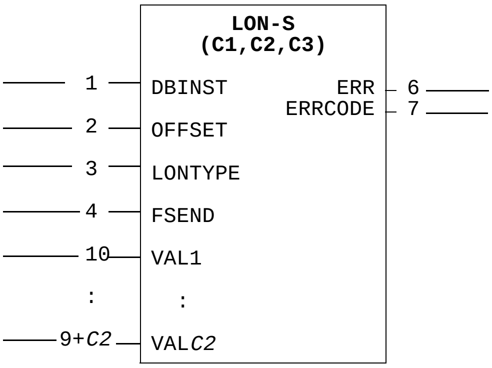
SNVT nr	Name	Conversions defined by LONTYPE	Range	Offset	R.f.	Length	Notes
244	NV_voltage_report (NV_volt_report)	FP	-1E38.....1E38	0	1	14	Voltage
		FP	-1E38.....1E38	4	1		Voltage
		FP	-1E38.....1E38	8	1		Voltage
		UL	0....6553.5	12	10		Frequency Hertz
243	NV_power_report	FP	-1E38.....1E38	0	1	12	Active power
		FP	-1E38.....1E38	4	1		Reactive power
		FP	-1E38.....1E38	8	1		Power factor
247	NV_clock	UL	0.....3000	0	1	9	Year
		US	0.....12	2	1		Month
		US	0.....31	3	1		Day
		US	0.....23	4	1		Hour
		US	0.....59	5	1		Minute
		US	0.....59	6	1	9	Second
		UL	0.....99 999	7	1		0.01 millisecond (10µs)
248*	NV_warning	US	30.....100	0	1	1	milliseconds
250*	NV_motor_ctrl	US	0.....255	0	1	1	
251*	NV_motor_state	US	0.....255	0	1	1	

* Temporary numbers

Send Data to LONWORKS Network LON-S

Summary

LON-S is used for sending data to an output Network Variable, defined by a LONNVO data base element.



Call LON-S(C1,C2,C3)

Call Parameters

Parameter	Significance	Permissible values	Default value
C1	Input VALue terminals data type	A, B, I, IL, R, T	-
C2	Number of data VALues	1.......32	1
C3	If C1=A: Number of bytes in Array	0.......40	0

Terminal Description23

Number	Name	Type	Description
1	DBINST	II	**D**ata **B**ase **INST**ance; A parameter defining the instance number of related data base element (LONNVO), alternatively, a symbolic connection to the element. (For example =LONNV147)
2	OFFSET	II	**OFFSET** in bytes for the start of the wanted (first) data value in this NV
3	LONTYPE	IA2	**LON TYPE** according to conversion table below. This terminal decides the data conversion type. You enter the value for LONTYPE as a two character string for example D='FP'.
4	FSEND	IB	**F**orce **SEND**ing Send data even if data hasn't changed.
6	ERR	OB	**ERR**or. This output terminal is set if an error is found.
7	ERRCODE	OI	**ERR**or **CODE**. See section "Error codes".
10	VAL1	*C1*	First data **VAL**ue to be sent.
:	:	:	
9 + *C2*	VAL*C2*	*C1*	Last data **VAL**ue to be sent

Function

LON-S sends data to the Network Variable defined by data base element LONNVO pointed out by DBINST terminal.

The terminal OFFSET gives the address adjustment (in bytes) from the start of the Network Variable to where the data component starts.

Terminal LONTYPE gives together with call parameter C1 the wanted conversion procedure (one line in the conversion table) which settles both how many bytes are affected and the transformation procedure. If different data types are possible in the application software then the conversion table gives alternative values of PC element call parameter C1.

If data doesn't fit the corresponding data type size the least significant bits are always used first when reducing the size and zeroes are added in the most significant positions when filling up.

NOTE

More than one LON-S PC element can update the same data field area. There is no checking that the combination of DBINST, OFFSET and C2 gives non-overlapping positions in the data field.

If two or more LON-S elements are set to operate on the same data, these elements ought to be placed in the same cycle time since **only one LON-S is allowed to operate at an NV at a time**.

If a collision occurs the first element blocks the other.

Error Codes

The following error codes may appear on the ERRCODE terminal:

0. No error.

1. Data base element given by DBINST or LONDEV is not found.
2. Data base element marked 'Not active'.
3. Invalid connection to DB element.
4. Data base element of incorrect type.
5. Data base element error marked.
6. Data base element not implemented.
7. Data base element not in service.
9. Data base element is not configured correctly.
10. CI572/CI573 send buffer full. Three consecutive sending failures.
11. Offset plus data length exceeds the Network Variable length value.
12. No bytes in array.
13. Bit field conversion conditions not met.
14. Communication area occupied by another PC element.
16. No NV status information available, but values are updated
17. Bus error. CI572/CI573 not accessible.
21. NV Status: Passive
23. NV Status: Error
24. NV Status: Unconfigured

ERRCODE continue:

30. NV Status Error: No parameter configured
31. NV Status Error: Parameter not correct
32. NV Status Error: Data not accessible
33. NV Status Error: Invalid data
35. NV Status Error: Heart-beat time-out
36. NV Status Error: Run-time error
37. NV Status Error: Update failed. (Check bindings of the Network Variable).
56. Unknown kind of LONTYPE.
57. Illegal data conversion is defined.
58. For 58, see LON-R.
101-109. Errors 1-9 referring to DB element LONNVO.

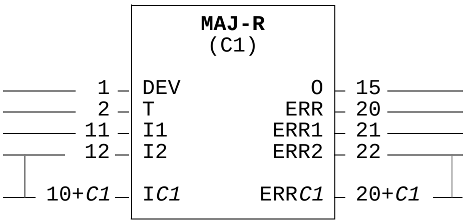
Conversion of data between LONWORKS and AMPL data types

See PC element LON-R.

Majority Selector MAJ-R

Summary

MAJ-R (**MAJ**ority selector **- R**eal) is used to calculate the majority value of 3 or 4 real numbers. The majority value and mean value are normally the same, but if an input deviates by more than a specific value, this input is excluded from the mean value calculation. If 2 or more signals are deviant, majority selection cannot be performed and the mean value is calculated from the values of all of the signals.



PC element MAJ-R

Call MAJ-R (C1)

Parameter	Significance	Permissible values
C1	Number of inputs	3 - 4

Terminal Description

No	Name	Type	Description
1	DEV	IR	**DEV**iation. Input which gives the maximum permissible deviation between the calculated mean value and one of the inputs.
2	T	IT	**T**ime. Input for the maximum time during which a signal may deviate from the mean value by more than DEV before an error signal is generated
11	I1	IR	**I**nput **1**.
12	I2	IR	**I**nput **2**
: 10+*C1*	I*C1*	IR	**I**nput ***C1***.
15	O	OR	**O**utput. Output for the mean value of the signals which does not deviate by more than the value at the input DEV.
20	ERR	OB	**ERR**or. Output which is set (to 1) when a majority selection is not possible.
21	ERR1	OB	**ERR**or **1**. Output which is set when input 1 has departed from the mean value by more than DEV for a time longer than T.
22	ERR2	OB	**ERR**or **2**. Output which is set when input 2 has departed from the mean value by more than DEV for a time longer than T.
: 20+*C1*	ERR*C1*	OB	**ERR**or ***C1***. Output which is set when input C1 has departed from the mean value by more than DEV for a time longer than T.

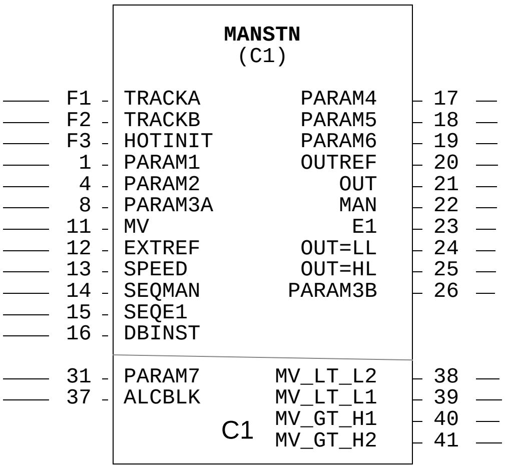
Function

The majority selector calculates the mean value of the inputs. If any input departs from the calculated mean value by more than DEV, that input is excluded from the mean value calculation. The corresponding ERR output is then set. If more than one input deviates by more than DEV, the mean value of all of the input values is calculated and the output ERR is set.

Manual Station MANSTN

Summary

The PC element MANSTN forms part of the function unit MANSTN (**MAN**ual **ST**atio**N**). The unit is primarily used for manual control of continuously presettable control objects or for manual references to other control equipment. It is used in traditional instrumentation when a VDU based operator station is required. Further information can be found in the manual Functional Units part 4, PIDCON, RATIOSTN, MANSTN.



PC element MANSTN

Call MANSTN (C1)

The element consists of one basic part that may be extended with an additional function.

Parameter	Significance	Permissible values
C1	Selection of measured value alarm	0 - 1

Terminal Description

No	Name	Type	Description
F1	TRACKA	II	**TRACK**ing **A**. Function parameter for tracking A.
F2	TRACKB	II	**TRACK**ing **B**. Function parameter for tracking B.
F3	HOTINIT	II	**HOT INIT**. Function parameter for warm start.
1	PARAM1	II	**PARAM**eter **1**. Connection to the data base element (PARAM1).
4	PARAM2	IA16	**PARAM**eter **2**. Connection to the data base element (PARAM2).
8	PARAM3A	II	**PARAM**eter **3A**. Connection to the data base element (PARAM3).
11	MV	IR	**M**easured **V**alue. Input for measured value.
12	EXTREF	IR	**EXT**ernal **REF**erence. Input for external reference.
13	SPEED	IR	Input for maximum change rate for MAN output signal (engineering unit/second).
14	SEQMAN	IB	**SEQ**uential **MAN**ual. Pulse input for ordering manual operation mode.
15	SEQE1	IB	**SEQ**uential **E1**. Pulse input for ordering operation mode E1.

Terminal Description

No	Name	Type	Description
16	DBINST	II	**D**ata **B**ase **INST**ance. An operational parameter or a constant, defining the instance number of the data base element (to be defined in its header). Alternatively, a connection to the data base element.
17	PARAM4	OI	**PARAM**eter **4**. Connection to the data base element (PARAM4).
18	PARAM5	OI	**PARAM**eter **5**. Connection to the data base element (PARAM5).
19	PARAM6	OA16	**PARAM**eter **6**. Connection to the data base element (PARAM6).
20	OUTREF	OR	**OUT**put **REF**erence. Reference for manual output signal.
21	OUT	OR	**OUT**put. Output signal.
22	MAN	OB	**MAN**ual. Output for indication of operation mode. This output is set to 1 in operation mode MAN.
23	E1	OB	**E1**. Output for indication of operation mode. This output is set to 1 in operation mode E1.
24	OUT=LL	OB	**OUT**put**=L**ow **L**imit. Output for indication of output signal limiting. This output is set to 1 when the output signal has been limited to a lower limit value.
25	OUT=HL	OB	**OUT**put**=H**igh **L**imit. Output for indication of output signal limiting. This output is set to 1 when the output signal has been limited to a higher limit value.
26	PARAM3B	OI	**PARAM**eter **3B**. Connection to the data base element (PARAM3).
31(1)	PARAM7	IA24	**PARAM**eter **7**. Connection to the data base element (PARAM7).
37(1)	ALCBLK	IB	**AL**arm **C**ontrol **BL**oc**K**. Input which, when set to 1, blocks the limit value alarm for the measured value (MV).
38(1)	MV_LT_L2	OB	**M**easured **V**alue_**L**ess **T**han_**L**ow **2**. Indication of measured value limit. This output is set to 1 when the measured value (MV) falls below alarm limit L2.
39(1)	MV_LT_L1	OB	**M**easured **V**alue_**L**ess **T**han_**L**ow **1**. Indication of measured value limit. This output is set to 1 when the measured value (MV) falls below alarm limit L1.
40(1)	MV_GT_H1	OB	**M**easured **V**alue_**G**reater **T**han_**H**igh **1**. Indication of measured value limit. This output is set to 1 when the measured value (MV) exceeds alarm limit H1.
41(1)	MV_GT_H2	OB	**M**easured **V**alue_**G**reater **T**han_**H**igh **2**. Indication of measured value limit. This output is set to 1 when the measured value (MV) exceeds alarm limit H2.

(1) These connections are available only if the call parameter C1 is set to 1**.**

Selection of parameters for the tracking function of MANOUT

Track A	Track B	Description
0	0	No tracking
1	0	MANOUT is not affected. Changeover to MAN takes place by OUT taking on the measured value, MV. Thereafter OUT is ramped to equal MANOUT with a speed that is determined by SPEED. ”Tracking” takes place in the operating mode E1.
2	0	MANOUT is not affected. Changeover to MAN takes place bumplessly by OUT retaining the value of EXTREF. Thereafter OUT is ramped to equal MANOUT with a speed that is determined by SPEED. ”Tracking” takes place in the operating mode E1.
1	1	MANOUT tracks the measured value, MV. Changeover to MAN takes place by MANOUT taking on the measured value, MV. Tracking takes place in the operating mode E1.
2	1	MANOUT tracks the external reference EXTREF. Changeover to MAN takes place both without bumps and changes by MANOUT taking on the value of the external reference EXTREF. Tracking takes place in the operating mode E1.

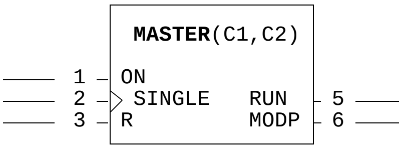
HOTINIT

HOTINIT	Function with warm start Operation mode	Output signal
0	unchanged	unchanged
1	MAN	unchanged
2	unchanged	unchanged
“unchanged” means: no change in relation to the state in effect before the event that caused the warm start.		

Master Header MASTER

Summary

The master header MASTER is used for the execution control of subordinate slave modules.



PC element MASTER

Call MASTER (C1, C2)

Parameter	Significance	Permissible values
C1	Cyclicity in ms	Permissible values are determined by the possible cycle times of the relevant PC system. These times are specified in the manual for the system concerned.
C2	Place in the cycle time table	Permissible values are determined by the relation to other execution units with the same cycle time. The max value is determined by the number of possible places with the same cycle time. The system selects a free place with the lowest number if no value is specified at entry.

Terminal Description

No	Name	Type	Description
1	ON	IB	Control input which is to be set (to 1) with normal execution.
2	SINGLE	IB	If input ON and R are 0 the elements in the slave modules can be executed once by setting SINGLE to 1.
3	R	IB	**R**eset. Input for clearing of the slave modules.
6	MODP	OB	**MOD**ify **P**ermission. Output that indicates if the master and slave modules may be modified from service aids.

Function

MASTER is a superior execution controlling element for a number of slave modules. How often the master and associated slave modules are to be executed is determined with the call parameter C1, and the place in the cycle time table is determined with the call parameter C2. The sequence in which execution units with the same cycle time are to be executed is based on the place in the cycle time table. When and how the slave modules are to be executed is determined by the control inputs ON, SINGLE and R. The master is always executed regardless of the state of these inputs.

Normal Execution

For normal execution the input ON must be set to 1. The master module and associated slave modules are then executed as determined by the call parameters C1 and C2. The order in which the subordinate slave modules are to be executed is determined by their order in the documentation.

If the ON input is reset after execution of the slave modules, the calculated data remains until the next execution of the slave modules. The ON input overrides the SINGLE input (if ON is set to 1. SINGLE has no effect. The slave modules are executed successively in three steps:

1. Entry of variables from the data base and from modules with higher priority.
2. Element execution.
3. Reading of variables to the data base and to modules with higher priority.

SINGLE Execution

If the SINGLE input is set when the ON input is reset, the slave modules are executed once as described above. The calculated data remains until the next execution of the master and slave modules.

Clearing

If the input R is set, the slave modules are executed in the reset mode. This means that all outputs of the elements in the slave modules are given initial values, which in most cases are the 0-value of the data concerned.

RUN

The output RUN is set only during normal execution, that is, ON is set or SINGLE execution is performed.

MODP

The master header contains a built in modification check, MODP. If the right to modify is deactivated, MODP=0, the possibility of entering changes in the functions of the master and slave modules from service aids is blocked. Blocking can however be introduced and operational parameters can be changed. The changing rights are activated and deactivated from service aids with special commands.

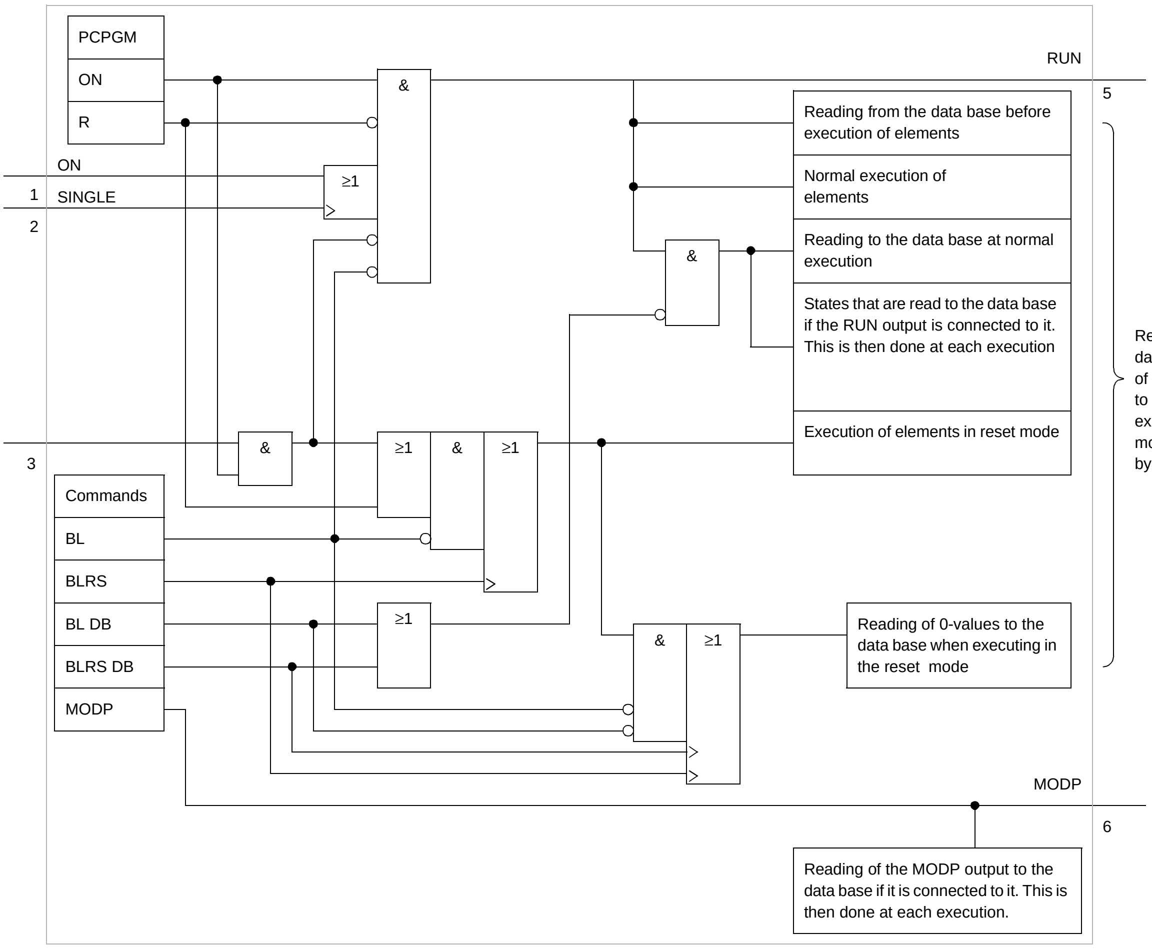
Blocking

The execution of the slave modules can be blocked in different ways from a service aid. All subordinate slave modules are blocked at the same time.

- The command BL<module identity>(**BL**ock) blocks all execution of the slave modules including data exchange with the data base.
- The command BL DB,<module identity>(**BL**ock **D**ata **B**ase) blocks reading to the data base of the data base variables which occur in the slave modules.
- The command BLRS<module identity>(**BL**ock and **R**e**S**et) first clears the slave modules and data to the data base after which the execution of the slave modules is blocked.
- The command BLRS DB,<module identity>(**BL**ock and **R**e**S**et **D**ata **B**ase) first clears output data to the data base for the slave modules after which all further reading is blocked.
- The command DBL<module identity>(**D**e**BL**ock) is used to cancel blockings.

Effects from PCPGM

The ON and R inputs and the blocking functions on the program header override the inputs on the master module header. The input ON on PCPGM must be set to permit normal execution of master and slave modules.

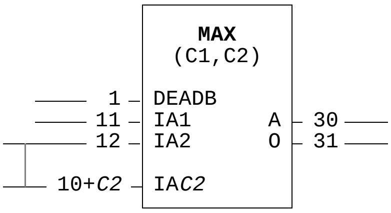

PCPGM
ON
R
RUN
&
5
Reading from the data base before execution of elements
Normal execution of elements
Reading to the data base at normal execution
States that are read to the data base if the RUN output is connected to it. This is then done at each execution
Execution of elements in reset mode
ON
1
SINGLE
2
≥1
3
Commands
BL
BLRS
BL DB
BLRS DB
MODP
Reading of 0-values to the data base when executing in the reset mode
Reading from the data base, execution of elements, reading to the data base and execution in reset mode is done slave by slave
MODP
6
Reading of the MODP output to the data base if it is connected to it. This is then done at each execution.

Function diagram

Maximum Selector MAX

Summary

MAX (**MAX**imum selector) is used to select the largest value of an optional number of integers or real numbers.



PC element MAX

Call MAX (C1, C2)

Parameter	Significance	Permissible values
C1	Data type	I, IL, R
C2	Number of inputs	2 - 19

Terminal Description

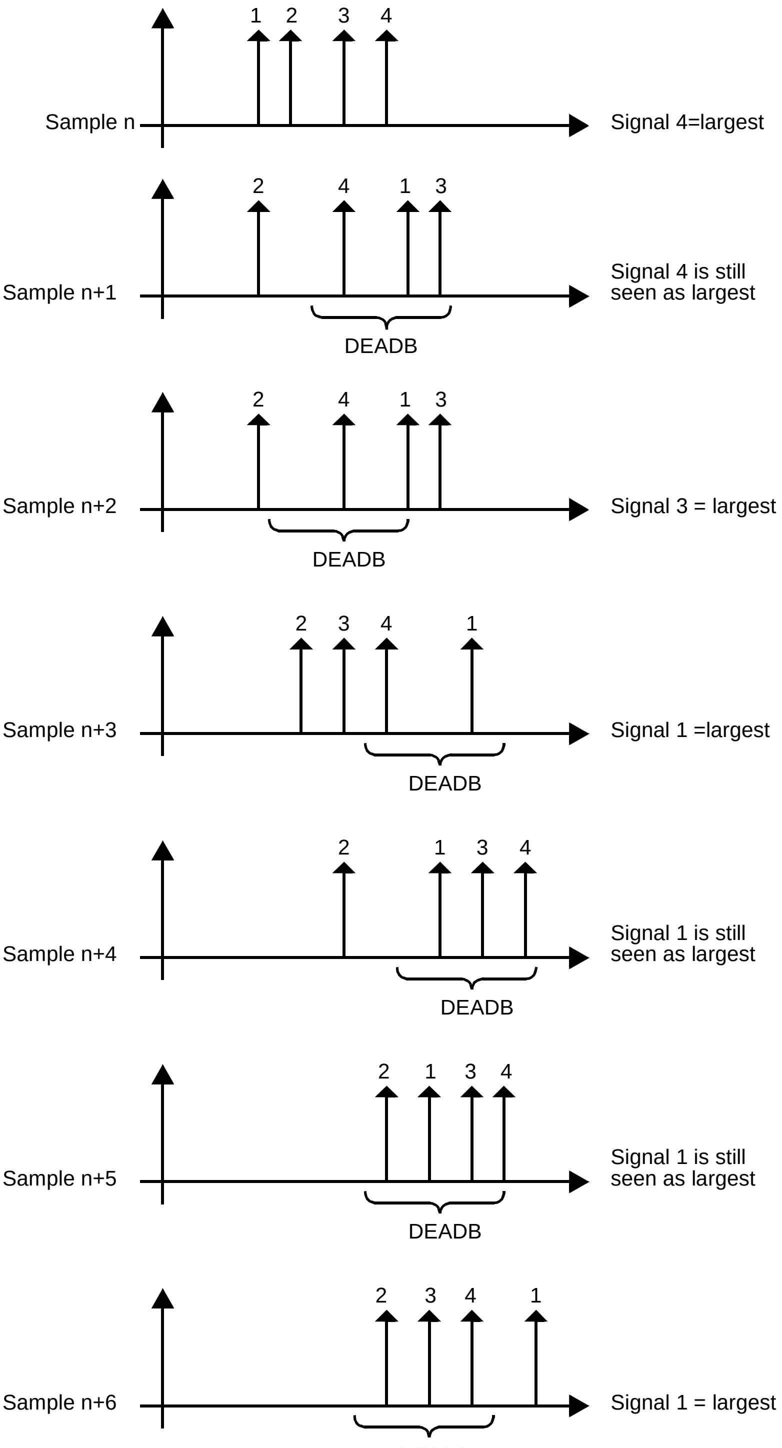
No	Name	Type	Description
1	DEADB	I*C1*	**DEADB**and. Input for deadband.
11	IA1	I*C1*	**I**nput **A**ddress **1**. Input the value of which is compared with the other inputs.
12	IA2	I*C1*	**I**nput **A**ddress **2**. Input the value of which is compared with the other inputs.
'			
10+*C2*	IA*C2*	I*C1*	**I**nput **A**ddress ***C2***. Input the value of which is compared with the other inputs.
30	A	OI	**A**ddress. Output for the number of the input having the greatest value.
31	O	O*C1*	**O**utput. Output for the greatest value.

Function

The values at the inputs IA1 - IA*C2* are compared and the greatest value is obtained at the output O. The number of the input with the greatest value is obtained at the output A. If the two largest signal values are equal when the element is executed the first time, the signal with the lowest connection number is selected as the greatest.

Deadband

The deadband specified at the input DEADB is symmetrical around the value of the greatest input. The upper and lower deadband limits are calculated from the value for the largest input in the preceding sample. To prevent rapid changes at the output A, the value at A is retained until the value at the corresponding input is less than the calculated lower deadband limit or until one of the other inputs exceeds the upper deadband limit.

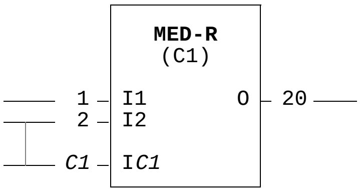


Example of function with deadband

Median Selector MED-R

Summary

MED-R (**MED**ian **- R**eal) is used to calculate the median value of an optional number of real numbers.



PC element MED-R

Call MED-R (C1)

Parameter	Significance	Permissible values
C1	Number of inputs	1 - 19

Terminal Description

No	Name	Type	Description
1	I1	IR	**I**nput **1**. Input for the first real value.
2	I2	IR	**I**nput **2**. Input for the second real value.
: *C1*	I*C1*	IR	**I**nput ***C1***. Input for the last real value.
20	O	OR	**O**utput. Output for the median value of the inputs.

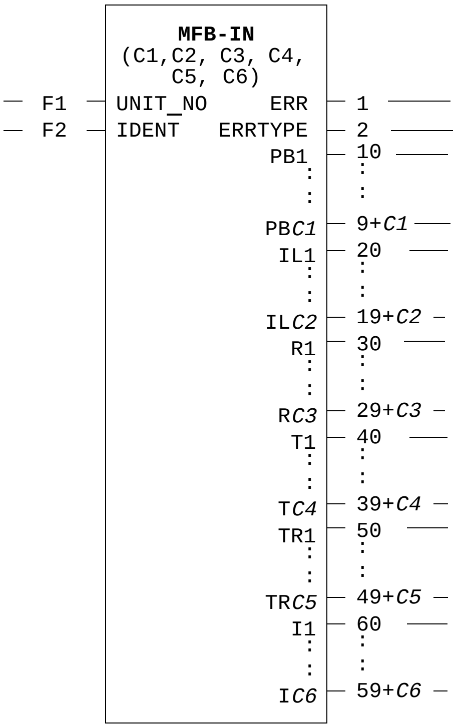
Function

The element calculates the median value of the inputs. If the number of inputs is odd, as many of the highest as of the lowest values are eliminated so that only one input remains. The value at this input is stored at the output. If the number of inputs is even, the same procedure is followed except that 2 inputs remain. The arithmetic mean of the values of these inputs is stored at the output.

Data from Advant Controller 110 MFB-IN

Summary

MFB-IN is used for transferring data to Advant Controller 410/450 from an Advant Controller 110 (AC 110) (or a MasterPiece 90) connected via MasterFieldbus.



PC element MFB-IN

Call MFB-IN (C1, C2, C3, C4, C5, C6)

Parameter	Significance	Permissible values
C1	Number of packed booleans	0 - 9 Default 0
C2	Number of IL values	0 - 9 Default 0
C3	Number of Real values	0 - 9 Default 0
C4	Number of Time values	0 - 9 Default 0
C5	Number of TimeReal values	0 - 9 Default 0
C6	Number of I values	0 - 19 Default 0
2×C1+2×C2+2×C3+2×C4+2×C5+C6≤19 C1+C2+C3+C4+C5+C6≥1		

Terminal Description

No	Name	Type	Description
F1	UNIT_NO	FI	**N**umber of the **Unit**. Permitted values are 100-115, 200-215, 300-315, 400-415, that is, 100xbus+Jumper address to first node area.
F2	IDENT	FI	**IDENT**ification. Range 1 - 16. Used to select node area of the MFb.
1	ERR	OB	**ERR**or. Indicate error at initiating at communication error.
2	ERRTYPE	OI	ERRor TYPE. Error code.
10	PB1	OIL	**P**acked **B**oolean **1**. 32 booleans packed as a 32 bit integer.
: 9+*C1*	PB*C1*	OIL	**P**acked **B**oolean **C1**. 32 booleans packed as a 32 bits integer.
20	IL1	OIL	**I**nteger **L**ong **1** from the Advant Controller 110.
: 19+*C2*	IL*C2*	OIL	**I**nter **L**ong **C2** from the Advant Controller 110.
30	R1	OR	**R**eal **1** from the Advant Controller 110.
: 29+*C3*	R*C3*	OR	**R**eal **C3** from the Advant Controller 110.
40	T1	OT	**T**ime **1** from the Advant Controller 110.
: 39+*C4*	T*C4*	OT	**T**ime **C4** from the Advant Controller 110.
50	TR1	OTR	**T**ime**R**eal **1** from the Advant Controller 110.
: 49+*C5*	TR*C5*	OTR	**T**ime**R**eal **C5** from the Advant Controller 110.
60	I1	OI	**I**nteger **1** from the Advant Controller 110.
: 59+*C6*	I*C6*	OI	**I**nteger **C6** from the Advant Controller 110.

Function

MFB-IN is used for transferring variables (orders) to the Advant Controller 410/450 from the Advant Controller 110 via MasterFieldbus. You need the data base element PX for startup of communication. See the manual ”Data Base Elements”.

With help of calling parameters, the number and type of variables are choosed. Up to 19 variables, depending on data type, can be transferred in one element. The available data type is 32 bits packed boolean, 32 bits integer, 32 bits real, 32 bits time, 32 bits timereal and 16 bits integer.

Address Selection

One Advant Controller 110 can occupy up to 16 addresses of one MasterFieldbus. Each address corresponds to one physical node area. One MFB-IN is needed for each node area. The address to the first one is defined by the function parameter UNIT _NO. The node areas for one AdvantController 110 is defined by the function parameter IDENT.

Example: One AC110, occupying two node areas, is connected at bus 3, address 300 and 301. Another AdvantController 110, also occupying two node areas, is connected at bus 3, address 302 and 303.

The first AC110	The first MFB-IN:	UNIT_NO=300, IDENT=1
The first AC110	The second MFB-IN:	UNIT_NO=300, IDENT=2
The second AC110	The first MFB-IN:	UNIT_NO=302, IDENT=1
The second AC110	The second MFB-IN:	UNIT_NO=302, IDENT=2

Each minute the MFB-IN in the Advant Controller 110 transmits its calling parameters to the MFB-IN in the Advant Controller 410/450. If the calling parameters between the sending and the receiving element are not equal, both elements set the output terminal ERR to true and the error code on terminal ERRTYPE to –10.

Configuration Check

Each minute the MFB-IN in the Advant Controller 110 transmits its calling parameters to the MFB-IN in the Advant Controller 410/450. If the calling parameters between the sending and the receiving element are not equal, both elements set the output terminal ERR to true and the error code on terminal ERRTYPE to –10.

Fault Handling

If an error is detected during initiating or normal communication, the output ERR is set to TRUE. The error type is mirrored in output ERRTYPE. If several errors exist simultaneously, the error code that occurs first is shown and the next remaining error is shown when the first one disappears. Possible error codes are:

Error Codes

ERRTYPE	Error Description
0	OK.
–1	The data base element given by the parameter UNIT_NO is missing.
–2	The data base element given by the parameter UNIT_NO is not of type PX.
–3	The data base element given by the parameter UNIT_NO is not accepted. This may be caused by the parameter IMPL being 0 or that the bus number is erroneous.
4	The bus controller module is missing or faulty.
5	The CI 670 unit is missing on the bus.
6	Initiation is taking place. Remaining indication indicates system error.
7	Connected node is not a CI 670.
8	Internal error on the controller module.
–9	The result of the function parameters F1+F2 is greater than x16.
–10	Mismatch between data types defined by the calling parameters and the initiation information coming from the Advant Controller 110.
11	The PC element is executing in reset mode.

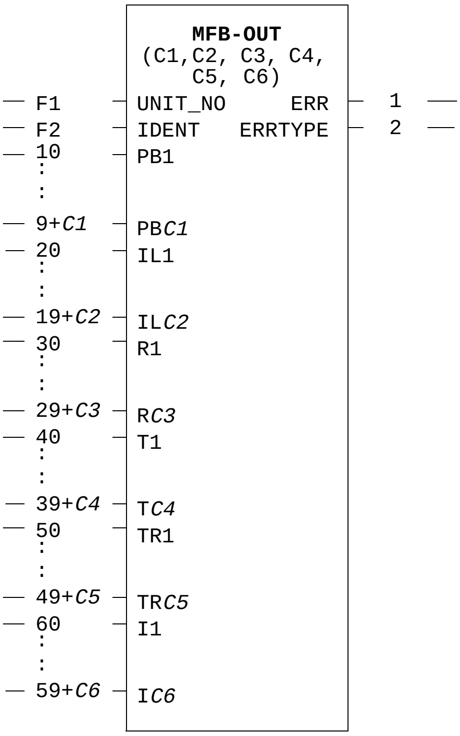
Error Codes

ERRTYPE	Error Description
12	Data from the Advant Controller 110 unit is not correct.
13	Mismatch between data base type and function parameter F2.
–100	Internal data base error.
The negative error numbers indicate configuration errors	

Data to Advant Controller 110 MFB-OUT

Summary

MFB-OUT is used for transferring data from Advant Controller 410/450 to an Advant Controller 110 (AC 110) (or a MasterPiece 90) connected via MasterFieldbus.



PC element MFB-OUT

Call MFB-OUT (C1, C2, C3, C4, C5, C6)

Parameter	Significance	Permissible values
C1	Number of packed booleans	0 - 9 Default 0
C2	Number of IL values	0 - 9 Default 0
C3	Number of Real values	0 - 9 Default 0
C4	Number of Time values	0 - 9 Default 0
C5	Number of TimeReal values	0 - 9 Default 0
C6	Number of I values	0 - 19 Default 0
2×C1+2×C2+2×C3+2×C4+2×C5+C6≤19 C1+C2+C3+C4+C5+C6≥1		

Terminal Description

No	Name	Type	Description
F1	UNIT_NO	FI	**N**umber of the **Unit**. Permitted values are 100-115, 200-215, 300-315, 400-415, that is, 100xbus+Jumper address to the first used MFb unit.
F2	IDENT	FI	**IDENT**ification. Range 1 - 16. Used to select node area of the MFb.
1	ERR	OB	**ERR**or. Indicates error at initiating at communication error.
2	ERRTYPE	OI	**ERR**or **TYPE**. Error code.
10	PB1	IIL	**P**acked **B**oolean 1. 32 booleans packed as a 32 bit integer.
: 9+*C1*	PB*C1*	IIL	**P**acked **B**oolean **C1**. 32 booleans packed as a 32 bits integer.
20	OL1	IIL	**I**nteger **L**ong **1** to the Advant Controller 110.
: 19+*C2*	IL*C2*	IIL	**I**nter **L**ong **C2** to the Advant Controller 110.
30	R1	IR	**R**eal **1** to the Advant Controller 110.
: 29+*C3*	R*C3*	IR	**R**eal **C3** to the Advant Controller 110.
40	T1	IT	**T**ime **1** to the Advant Controller 110.
: 39+*C4*	T*C4*	IT	**T**ime **C4** to the Advant Controller 110.
50	TR1	ITR	**T**ime**R**eal **1** to the Advant Controller 110.
: 59+*C6*	I*C6*	OI	**I**nteger **C6** to the Advant Controller 110.

Function

PC element MFB-OUT is used for transferring variables (orders) from the Advant Controller 400 system to the Advant Controller 110 connected via MasterFieldbus. With help of calling parameters, the number and type of variables are chosen. Up to 19 variables, depending on data type, can be transferred in one element. The available data type is 32 bits packed booleans, 32 bits integer, 32 bits real, 32 bits time, 32 bits timereal and 16 bits integer.

Address Selection

One Advant Controller 110 (AC110) can occupy up to 16 addresses of one MasterFieldbus. Each address corresponds to one physical node area. One MFB-OUT PC element is needed for each node area. The address to the first node area is defined by the function parameter UNIT _NO. The node areas for one AC110 is defined by the function parameter IDENT.

Example: One AC110, occupying two node areas, is connected at bus 3, address 300 and 301. Another AC110, also occupying two node areas, is connected at bus 3, address 302 and 303.:

The first AC110	The first MFB-OUT:	UNIT_NO=300, IDENT=1
The first AC110	The second MFB-OUT:	UNIT_NO=300, IDENT=2
The second AC110	The first MFB-OUT:	UNIT_NO=302, IDENT=1
The second AC110	The second MFB-OUT:	UNIT_NO=302, IDENT=2

Configuration Check

To be able to do a configuration check, the element in Advant Controller 400 transfers its calling parameters to the Advant Controller 110 once every minute. If the calling parameters are not equal in the Advant Controller 410/450 (MFB-OUT) and Advant Controller 110, the error code –10 is set in both elements.

Fault Handling

If an error is detected at initiating or during communication the output ERR is set to TRUE. The error type is mirrored in output ERRTYPE. If several exist simultaneously, the error code that occurs first is shown and the next remaining error is shown when the first one goes away. Possible error codes are:

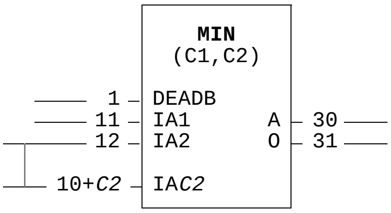
Error codes

ERRTYPE	Error description
0	OK.
–1	The data base element given by the parameter UNIT_NO is missing.
–2	The data base element given by the parameter UNIT_NO is not of type PX.
–3	The data base element given by the parameter UNIT_NO is not accepted. This may be caused by the parameter IMPL being 0 or that the bus number is erroneous.
4	The bus controller module is missing or faulty.
5	The CI 670 unit is missing on the bus.
6	Initiation is taking place. Remaining indication indicates system error.
7	Connected node is not a CI 670.
8	Internal error on the controller module.
–9	The result of the function parameters F1+F2 is greater than x16.
–10	The calling parameters for the MFB-OUT element is not the same as in the receiving element.
11	The PC element is executing in reset mode, and the value zero is sent to the CI670-unit.
13	Mismatch between data base type and function parameter F2.
–100	Internal data base error.

Minimum Selector MIN

Summary

MIN (**MIN**imum selector) is used to select the lowest value of an optional number of integers or real numbers.



PC element MIN

Call MIN (C1, C2)

Parameter	Significance	Permissible values
C1	Data Type	I, IL, R
C2	Number of input	2 - 19

Terminal Description

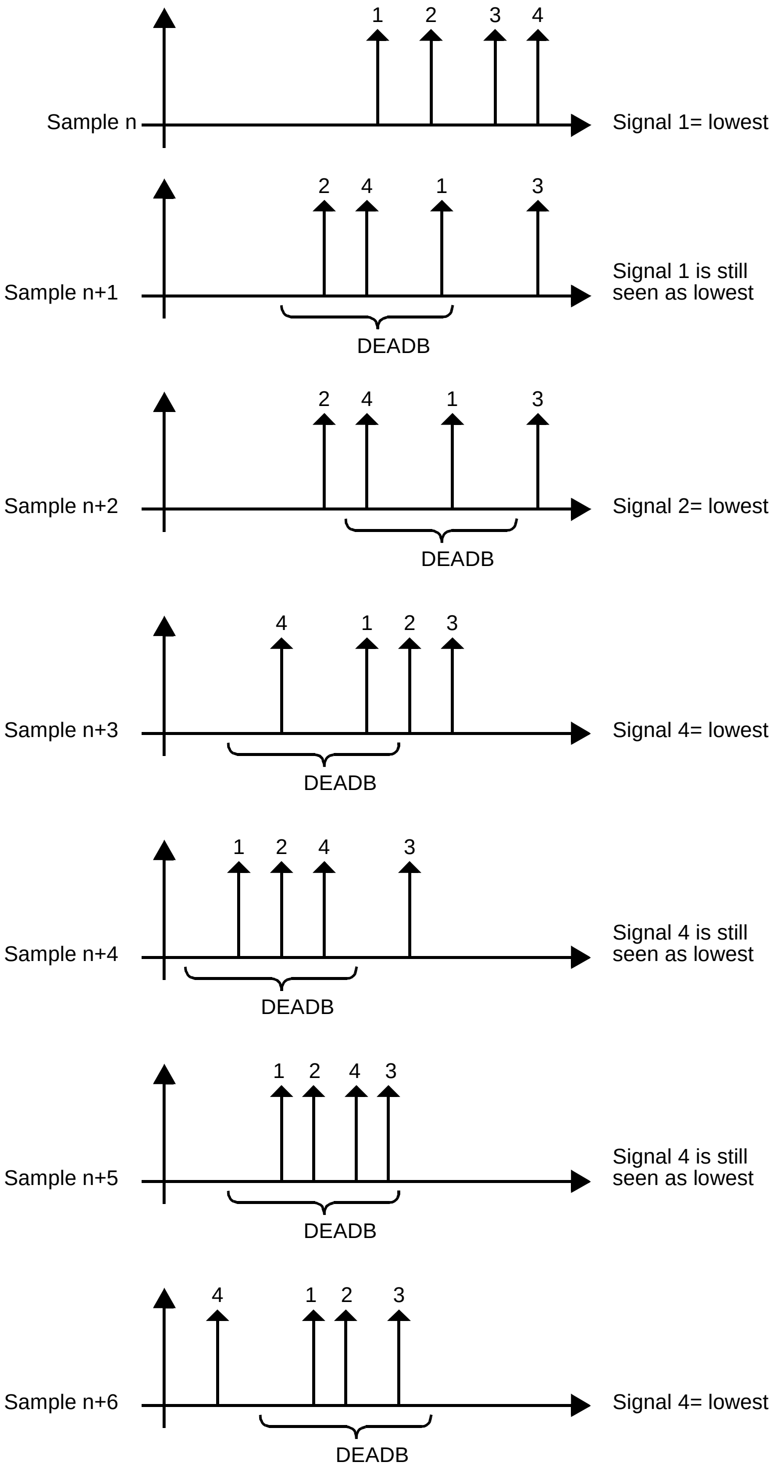
No	Name	Type	Description
1	DEADB	I*C1*	**DEADB**and. Input for deadband.
11	IA1	I*C1*	**I**nput **A**ddress **1**. Input the value of which is compared with the other inputs.
12	IA2	I*C1*	**I**nput **A**ddress **2**. Input the value of which is compared with the other inputs.
: 10+*C2*	IA*C2*	I*C1*	**I**nput **A**ddress ***C2***. Input the value of which is compared with the other inputs.
30	A	OI	**A**ddress. Output for the number of the input having the lowest value.
31	O	I*C1*	**O**utput. Output for the least value.

Function

The values at the inputs IA1 - IA*C2* are compared and the lowest value is obtained at the output O. The number of the input with the lowest value is obtained at the output A. If the two smallest signal values are equal when the element is executed the first time, the signal with the lowest connection number is selected as the smallest.

Deadband

The deadband specified at the input DEADB is symmetrical around the value of the least input. The upper and lower deadband limits are calculated from the value for the lowest input in the preceding sample. To prevent rapid changes at the output A, the value at A is retained until the value at the corresponding input exceeds the calculated upper deadband limit or until one of the other inputs falls below the lower deadband limit.

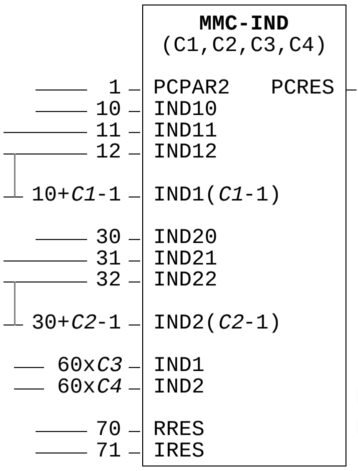


Example of function with deadband

Operator Station Indications MMC-IND

Summary

MMC-IND is used with the data base element MMCX, object handler, display elements of Advant Station and alarm and event handling. This enables the PC programmer to send indications and measured values from arbitrary PC programs as well as generate alarms and events to the operator.
The element can handle up to 32 indications, one real measured value and one integer measured value.
The indications are divided into two groups (IND1 and IND2) with regard to alarm and event handling.



PC element
MMC-IND

Call MMC-IND (C1, C2, C3, C4)

Parameter	Significance	Permissible values
C1	Number of inputs for IND1 indications.	0 - 16
C2	Number of inputs for IND2 indications.	0 - 16
C3[1]	Inputs used for IND1. C3=0: Indications via inputs 10 - 25. C3=1: Indications via input 60.	0 - 1
C4[2]	Inputs used for IND2. C4=0: Indications via inputs 30 - 45. C4=1: Indications via input 61.	0 - 1

(1) If C3 is set to 1, C1 must be 0.
(2) If C4 is set to 1, C2 must be 0.

Terminal Description

No	Name	Type	Description
1	PCPAR2	IA10	**PC PAR**ameter **2**. Parameters from the MMCX data base element, terminal PC_PARM2
5	PCRES	OA12	**PC RES**ult. Result to the MMCX data base element, terminal PC_RES
10	IND10	IB	**IND**ication **1** input **0**. Input for the first indication within IND1.
11	IND11	IB	**IND**ication **1** input **1**. Input for the second indication within IND1.
. 10+*C1–1*	IND1(*C1–1*)	IB	**IND**ication **1** input (*C1–1*). Input for (*C1–1*):st indication within IND1.
30	IND20	IB	**IND**ication **2** input **0**. Input for the first indication within IND2
31	IND21	IB	**IND**ication **2** input **1**. Input for the second indication within IND2.
. 30+*C2–1*	IND2(*C2–1*)	IB	**IND**ication **2** input (*C2–1*). Input for (*C2–1*):st indication within IND2.

Terminal Description

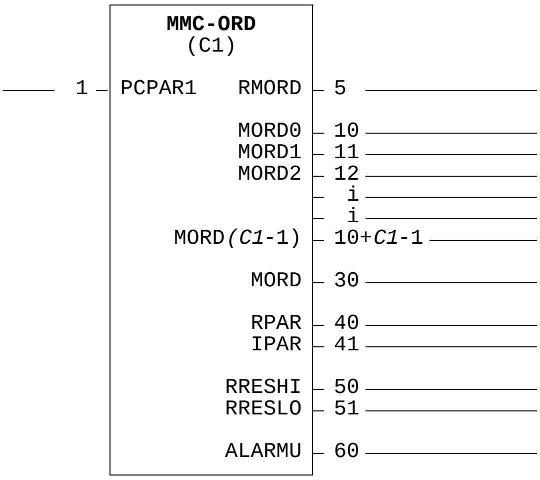
No	Name	Type	Description
60	IND1	II	**IND**ication **1**. Input for IND1 indications in packed form.
61	IND2	II	**IND**ication **2**. Input for IND2 indications in packed form.
70	RRES	IR	**R**eal **RES**ult. Input for real measured value.
71	IRES	IIL	**I**nteger **RES**ult. Input for integer measured value.

Operator Station Orders MMC-ORD

Summary

MMC-ORD is used with the data base element MMCX, object handlers and a suitable operator dialogue to give the PC programmer the possibilities to send orders, set point values, limit values and information of unacknowledged alarms from the operator to the PC program. The element can handle 16 on/off orders, one real set point value and one integer set point value.

If the element is used with an MMC-IND element in the PC program for a certain object, it also has the capability of presenting information of whether unacknowledged alarms are present among the 32 indications of the MMC-IND element or not. MMC-ORD can also present high and low limits for the real measured value of the MMC-IND element.



PC element MMC-ORD

Call MMC-ORD (C1)

Parameter	Significance	Permissible values
C1	Number of on/off orders	1 - 16

Terminal Description

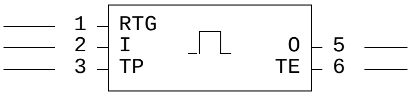
No	Name	Type	Description
1	PCPAR1	IA24	**PC PAR**ameter **1**. Parameters from the MMCX data base element, terminal PC_PARM1.
5	RMORD	OI	**R**eset **M**anual **ORD**er. Output for clearing on/off orders, terminal MORD.
10	MORD0	OB	**M**anual **ORD**er **0**. Output for the 1:st on/off order.
11	MORD1	OB	**M**anual **ORD**er **1**. Output for the 2:nd on/off order.
: 10+*C1*–1	MORD(*C1–1*)	OB	**M**anual **ORD**er *C1*–1. Output for the (*C1*–1):st on/off order.
30	MORD	OI	**M**anual **ORD**er. Output for the on/off orders in packed form.
40	RPAR	OR	**R**eal **PAR**ameter. Output for the real set point value.
41	IPAR	OIL	**I**nteger **PAR**ameter. Output for the integer set point value.
50	RRESHI	OR	**RES**ult **HI**gh limit. High limit for the real measured value(1).
51	RRESLO	OR	**R**eal **RES**ult **LO**w limit. Low limit for the real measured value (1).
60	ALARMU	OIL	**ALARM U**nacknowledged. Output for information about unacknowledged alarms among the indications, corresponding to IND10 to IND 215.

(1) Applies to the companion MMC-IND element.

Monostable MONO

Summary

The mono function MONO can be used for time limiting of operation of outputs and automatic functions, impulse extension, stall alarm function, etc.



PC element MONO

Call MONO

Terminal Description

No	Name	Type	Description
1	RTG	IB	**R**e**T**rig**G**. Input for selection of if the MONO function is to be retriggerable or not. If RTG is set, a retriggerable function is obtained.
2	I	IN	Input. Input for start of the MONO function when the input changes from 0 to 1.
3	TP	IT	**T**ime **P**ulse. Input for setting of pulse time.
5	O	OB	**O**utput. Output which is set when the MONO function starts and resets (to 0) when the time set has elapsed.
6	TE	OT	**T**ime **E**lapsed. Output for time which has elapsed when O is set. When the time set (TP) has elapsed TE will stop.

Function

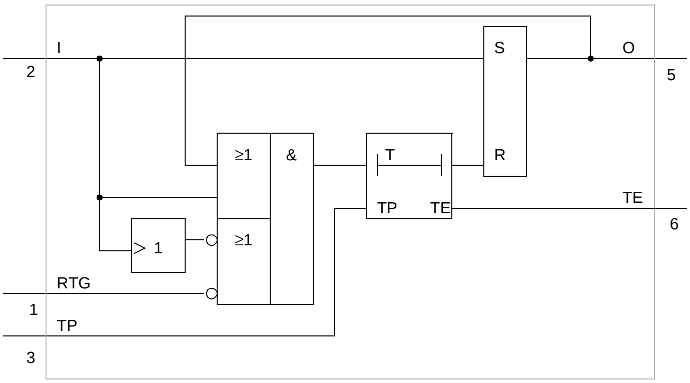
A memory is set when the input I is set. The output then goes to 1, see function diagram. When the time set in the timer has elapsed, the memory is cleared and the output O goes to 0.

MONO Function, Not Retriggerable

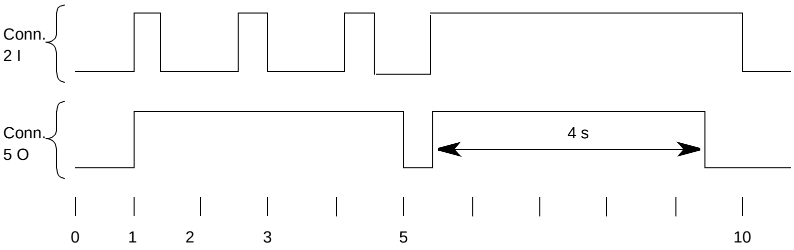
If the input RTG is reset, a MONO function which is not retriggerable is obtained. If a new pulse is obtained at the input I before the time set in the timer has elapsed, it does not affect the timer. Only when the time set has elapsed and the output O is reset can the MONO function be restarted by the input I going from 0 to 1.

MONO Function, Retriggerable

If RTG is set, a retriggerable MONO function is obtained, that is, the timer starts from 0 each time a new pulse is obtained at the input I. If a new pulse is obtained at the input I before the time set in the timer has elapsed, the timer will restart, that is, the MONO function is retriggerable.

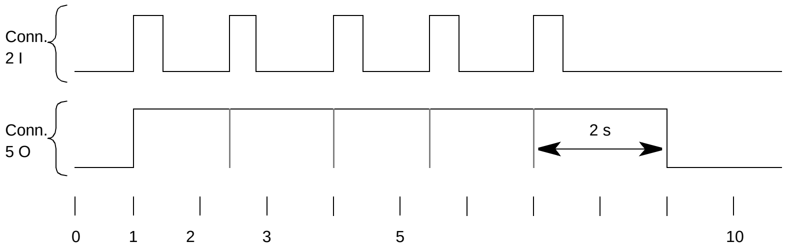


Function diagram



Conn.3 Time set=4 s
Conn.1 Reset

Time diagram MONO function, not retriggerable



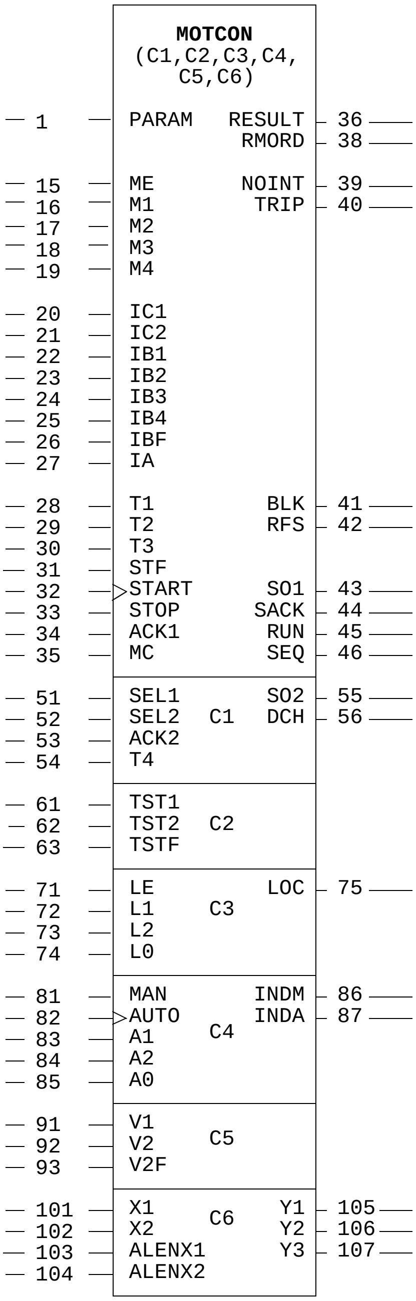
Conn.3 Time set=2 s
Conn.1 Set

Time diagram MONO function, retriggerable

Motor Controller MOTCON

Summary

The PC element MOTCON forms part of the functional unit MOTCON (MOTor CONtroller). The element is used with the data base element MOTCON for controlling motors.
More information can be found in the manual *Functional Units part 6, MOTCON, VALVECON*.



PC element MOTCON

Call MOTCON (C1, C2, C3, C4, C5, C6)

The element consists of one basic part which gives a complete motor controller in its basic form. This part may be complemented by optional extra functions with the call parameters C1–C6.

Parameter	Significance	Permissible values
C1	Selection of forward/backward (high/low) function.	0 - 1
C2	Selection of test run function.	0 - 1
C3	Selection of function for control from a local control panel.	0 - 1
C4	Selection of auto function (for example, run between limit positions).	0 - 1
C5	Selection of function for start interlocked sensors.	0 - 1
C6	Selection of function for external indication/alarm and external control via operator dialog.	0 - 1

Terminal Description

No	Name	Type	Description
1	PARAM	IA32	**PARAM**eter. Parameters from the object data base element MOTCON, terminal PC_PARAM
15	ME	IB	**M** circuit supervision **E**nable. Input for enabling alarm from control circuit.
16	M1	IB	Input for control voltage.
17	M2	IB	Input for bimetal relay.
18	M3	IB	Input for stop from motor location.
19	M4	IB	Input for safety supervision.
20	IC1	IB	**I**nterlock **C1**. Operation interlock (not deblockable).
21	IC2	IB	**I**nterlock **C2**. Operation interlock (not deblockable).
22	IB1	IB	**I**nterlock **B1**. Operation interlock (1) (2).
23	IB2	IB	**I**nterlock **B2**. Operation interlock(2).
24	IB3	IB	**I**nterlock **B3**. Start/operation interlock(1)(2).
25	IB4	IB	**I**nterlock **B4**. Start/operation interlock(2).
26	IBF	IB	**I**nterlock **B F**unction. Control input for IB3 and IB4. If IBF is set to 1, IB3 and IB4 work as start interlocks.
27	IA	IB	**I**nterlock A. Operation interlock(2).
28	T1	IT	**T**ime **1**. Input for pulse time for start order.
29	T2	IT	**T**ime **2**. Input for maximum delay time of contactor response.
30	T3	IT	**T**ime **3**. Input for time for delayed activation of V1 and V2. (This presupposes that call parameter C5=1.)
31	STF	IB	**ST**art **F**unction. Input for control function for direction change. Set STF=1. STF=0 is not implemented.
32	START	IB	Input for start order.

Terminal Description

No	Name	Type	Description
33	STOP	IB	Input for stop order. STOP=0 causes stop.
34	ACK1	IB	**ACK**nowledge **1**. Input for acknowledgement of SO1 (start acknowledgement).
35	MC	IR	**M**otor **C**urrent. Input for motor current.
36	RESULT	OA12	Output for results to the data base element MOTCON, terminal PC_RES.
38	RMORD	OI	**R**eset **M**anual **ORD**er. Output for clearing activation order in the object data base element MOTCON, terminal MORD.
39	NOINT	OB	**NO INT**erlock. Output which is set to 1 when the object is not interlocked and no remaining or unacknowledged faults exist.
40	TRIP	OB	Output which indicates an error in a control circuit, tripped supervision (V1, V2), contactor fault or external alarm (X1, X2).
41	BLK	OB	**BL**oc**K**ed. Output which is set to 1 when the point of control is STANDBY.
42	RFS	OB	**R**eady **F**or **S**tart. Output which is set to 1 when IC1zIC2 and IB1zIB4 are all 1 and (when the object is not in operation mode TEST) there are no unacknowledged or remaining faults.
43	SO1	OB	**S**tart **O**rder **1**. Output to contactor.
44	SACK	OB	**S**tart **ACK**nowledge. Output which indicates that the object has been started and the alarm is activated.
45	RUN	OB	Output for operation acknowledgement. (SO1 and ACK1 are set to 1.)
46	SEQ	OB	**SEQ**uence. Indication of point of control SEQUENCE.
51(3)	SEL1	IB	**SEL**ect start order **1**. Preselection of start order 1.(Does start the object.)
52(3)	SEL2	IB	**SEL**ect start order **2**. Preselection of start order 2. (Does not start the object.)
53(3)	ACK2	IB	**ACK**nowledge **2**. Input for acknowledgement of SO2 (start acknowledgement).
54(3)	T4	IT	**T**ime **4**. Input for minimum time for switching start orders (Change of direction).
55(3)	SO2	OB	**S**tart **O**rder **2**. Output to contactor.
56(3)	DCH	OB	**D**irection **CH**ange. Output which, when set to 1, indicates that a change of start order is in progress.
61(3)	TST1	IB	**T**est **ST**art **1**. Input for start order 1 from motor location.
62(3)	TST2	IB	**T**est **ST**art **2**. Input for start order 2 from motor location.
63(3)	TSTF	IB	**T**est **ST**art **F**unction. Input for self locking for TST1 and TST2. Self locking is obtained when TSTF is set to 1.
71(3)	LE	IB	**L**ocal **E**nable. Input for selection of point of control LOCAL.
72(3)	L1	IB	**L**ocal start order **1**. Input for local start order 1.
73(3)	L2	IB	**L**ocal start order **2**. Input for local start order 2.
74(3)	L0	IB	**L**ocal stop order. Input for local stop order. L0 is active low.
75(3)	LOC	OB	**LOC**al. Indication of point of control LOCAL.

Terminal Description

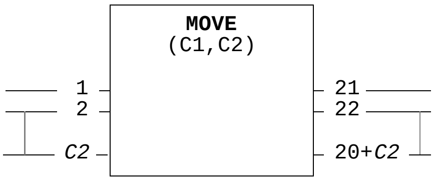
No	Name	Type	Description
81(3)	MAN<	IB	**MAN**ual. Input for switching to control mode MANUAL.
82(3)	AUTO	IB	**AUTO**matic. Input for switching to control mode AUTO. Enables the inputs A1, A2 and A0. Dynamic input.
83(3)	A1	IB	**A**uto **1**. Input for auto start order 1.
84(3)	A2	IB	**A**uto **2**. Input for auto start order 2.
85(3)	A0	IB	**A**uto **0**. Input for auto stop order 0.
86(3)	INDM	OB	**IND**ication **M**anual. Output for indication of control mode MANUAL.
87(3)	INDA	OB	**IND**ication **A**uto. Output for indication of control mode AUTO.
91(3)	V1	IB	Input for tripping (active low), to be connected to an operation monitor. The input is disabled for time T3 after start.
92(3)	V2	IB	Input for tripping (active low), to be connected to an operation monitor. The input is disabled for time T3 after start. It is further required that the input V2F is set to 1 for tripping to be possible.
93(3)	V2F	IB	**V2 F**unction. Input which must be set to 1 if tripping of V2 is to be possible.
101(3)	X1	IB	E**X**ternal alarm/indication **1**. Input for external alarm/indication. (IND107.)
102(3)	X2	IB	E**X**ternal alarm/indication **2**. Input for external alarm/indication. (IND111.)
103(3)	ALENX1	IB	**AL**arm **EN**able **X1**. Input which, when set to 1, gives alarm function at X1.
104(3)	ALENX2	IB	**AL**arm **EN**able **X2**. Input which, when set to 1, gives alarm function at X2.
105(3)	Y1	OB	Output controllable from dialog.
106(3)	Y2	OB	Output controllable from dialog.
107(3)	Y3	OB	Output controllable from dialog.

(1) May be disabled via operator dialog.
(2) Disabled in operation mode TEST.
(3) These terminals are selected with the call parameters C1-C6 respectively

Data Copying MOVE

Summary

MOVE copies the value at the input to the output and may be used to, for example, copy the value from one location in the data base to another.



PC element MOVE

Call MOVE (C1, C2)

Parameter	Significance	Permissible values
C1	Data type	B, I, IL, R, T, TR
C2	Number of inputs and outputs	1 - 19

Terminal Description

No	Name	Type	Description
1	–	I*C1*	Input for value 1.
2	–	I*C1*	Input for value 2.
: *C2*	–	I*C1*	Input for value C2.
21	–	O*C1*	Output for value 1.
22	–	O*C1*	Output for value 2.
: 20+*C2*	–	O*C1*	Output for value C2.

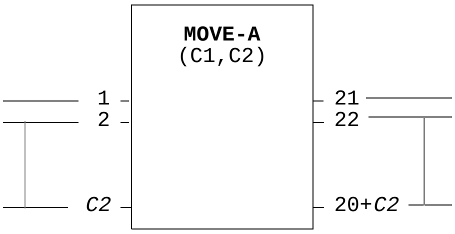
Function

The element copies the values at the inputs to the respective output. The type of the data to be moved is determined with the call parameter C1 and the number of values moved is determined with the call parameter C2.

Array Data Copying MOVE-A

Summary

MOVE-A (**MOVE-A**rray) copies the value at the input to the output and may be used to copy an array from one location in the data base to another.



PC element MOVE-A

Call MOVE-A (C1, C2)

Parameter	Significance	Permissible values
C1	Number of characters at the inputs and outputs.	1 - 80
C2	Number of inputs and outputs.	1 - 19

Terminal Description

No	Name	Type	Description
1	–	IA*C1*	Input for array 1.
2	–	IA*C1*	Input for array 2.
: *C2*	–	IA*C1*	Input for array C2.
21	–	OA*C1*	Output for array 1.
22	–	OA*C1*	Output for array 2.
: 20+*C2*	–	OA*C1*	Output for array 2.

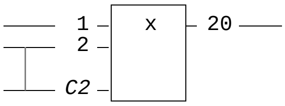
Function

The element copies the content of the arrays at the inputs to the respective output. The number of characters in the arrays is set with call parameter C1 and the number of arrays to be moved is set with call parameter C2.

Multiplier MUL

Summary

MUL (**MUL**tiplier) is used for multiplication of an optional number of integers or real numbers.



PC element MUL

Call MUL (C1, C2)

Parameter	Significance	Permissible values
C1	Data type	I, IL, R
C2	Number of inputs	2 - 19

Terminal Description

No	Name	Type	Description
1	–	I*C1*	Input for multiplicand
2	–	I*C1*	Input for multiplier
: *C2*	–	I*C1*	Input for multiplier
20	–	O*C1*	Output for product

Function

The values at the inputs 1 - *C2* are multiplied and the product is stored at the output 20.

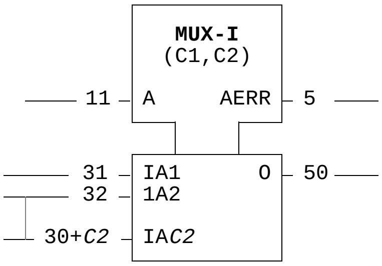
Overflow

If the maximum positive or negative values are exceeded, the output is limited to the highest or lowest representable value for the data type.

Multiplexer with Integer Address MUX-I

Summary

MUX-I (**MU**ltiple**X**er **-** with **I**nteger address) is used as a selector and has an optional number of inputs. The data type can be integers, real numbers, Boolean or time.



PC element MUX-I

Call MUX-I (C1, C2)

Parameter	Significance	Permissible values
C1	Data type	I, IL, R, B, T, TR
C2	Number of inputs	1 - 19

Terminal Description

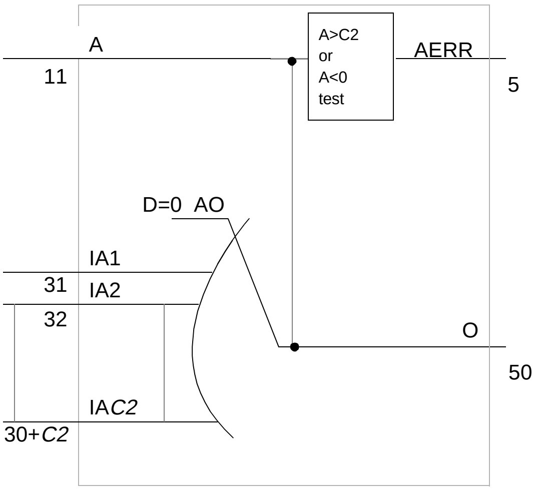
No	Name	Type	Description
5	AERR	OB	**A**ddress **ERR**or. Output which is set when the address value is larger than the number of inputs, or negative.
11	A	II	**A**ddress. Input for address value which specifies which input is to be connected to the output O. When A=0, the output is set to 0.
31	IA1	I*C1*	**I**nput **A**ddress **1**. The second input to the selector.
32	IA2	I*C1*	**I**nput **A**ddress **2**. The second input to the selector.
: 30+*C2*	IA*C2*	*C1*	**I**nput **A**ddress ***C2***. The last input to the selector.
50	O	O*C1*	**O**utput. Data output from the selector.

Function, Addressing

The input, the data value (IA1 - IA*C2*) of which is to be connected to the output O, is specified with an address (integer 1 - *C2*) at the input A. If the address is 0, the output terminal O is set to zero.

Supervision

The address A is monitored. If its value is negative or greater than the number of inputs, the error signal output AERR is set. The data value 0 is then obtained at the output.

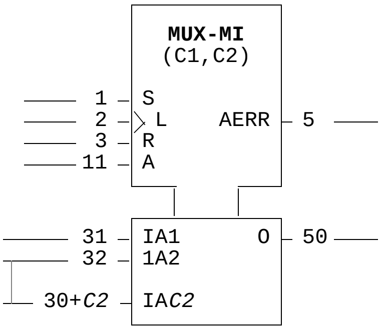


Function diagram

Multiplexer with Memory and Integer Address MUX-MI

Summary

MUX-MI (**MU**ltiple**X**er **-** with **M**emory and **I**nteger address) is used as a selector with memory function and has an optional number of inputs. The data type can be integers, real numbers, Boolean or time.



PC element MUX-MI

Call MUX-MI (C1, C2)

Parameter	Significance	Permissible values
C1	Data type	I, IL, R, B, T, TR
C2	Number of inputs	1 - 19

Terminal Description

No	Name	Type	Description
1	S	IB	**S**et. Input for loading of new values each time the element is executed. When the input S is reset 0, the last value loaded will remain at the outputs.
2	L	IB	**L**oad. Dynamic input for loading of the data addressed.
3	R	IB	**R**eset. Input for clearing of the outputs. This input overrides the S and L inputs.
5	AERR	OB	**A**ddress **ERR**or. Output which is set when the address value is greater than the number of inputs or is negative.
11	A	II	**A**ddress. Input for address value which specifies from which input data is to be loaded. When A=0, the output is reset to 0.
31	IA1	I*C1*	**I**nput **A**ddress **1**. The first input to the selector.
32	IA2	I*C1*	**I**nput **A**ddress **2**. The second input to the selector.
: 30+*C2*	A*C2*	I*C1*	**I**nput **A**ddress ***C2***. The last input to the selector.
50	O	O*C1*	**O**utput. Data output from the selector.

Function, Addressing

The input data value (IA1 - IA*C2*) that is to be stored at the output is specified with an address (integer 1 - *C2*) at the input A. If the address is 0, the data value 0 is stored.

Loading

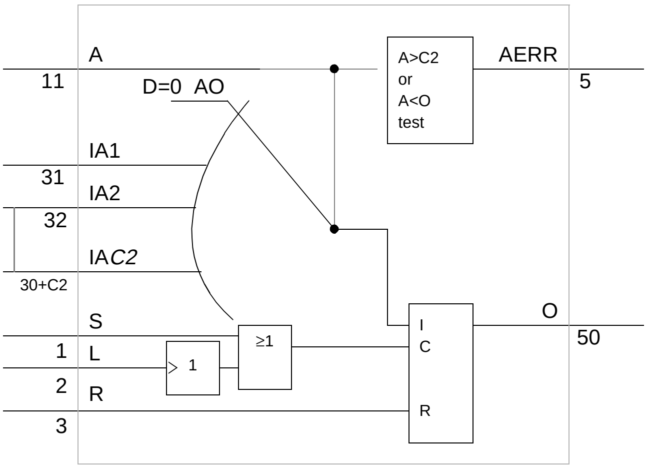
The value is loaded at the instant the input L is set to 1. If input S is set, new data is loaded each time the element is executed. When S is reset after having been set, the data loaded most recently remains until the element is once more executed with input S, L or R set. Input S overrides input L, that is, when S is set, L has no function.

Clearing

The input R clears the data output and prevents further storage of data.

Supervision

The address A is monitored and if its value is greater than the number of inputs or is negative, the error signal output AERR is set. The data value 0 is then stored at the output.

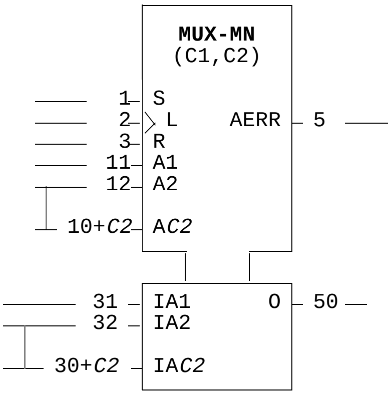


Function diagram

Multiplexer with Memory and 1 of N Address MUX-MN

Summary

MUX-MN (**MU**ltiple**X**er **-** with **M**emory and 1-of-**N** address) is used as a selector with memory function. MUX-MN has an optional number of inputs. The data type can be integers, real numbers, Boolean or time.



PC element MUX-MN

Call MUX-MN (C1, C2)

Parameter	Significance	Permissible values
C1	Data type	I, IL, R, B, T, TR
C2	Number of inputs	1 - 19

Terminal Description

No	Name	Type	Description
1	S	IB	**S**et. Input for loading of new values each time the element is executed. When the input S is reset, the last value loaded will remain at the outputs.
2	L	IB	**L**oad. Dynamic input for loading of the data addressed.
3	R	IB	**R**eset. Input for clearing of the outputs. This input overrides the S and L inputs.
5	AERR	OB	**A**ddress **ERR**or. Output which is set when two or more of the inputs A1 - A*C2* are set.
11	A1	IB	**A**ddress **1**. Input to store data from the input IA1 at the output O.
12	A2	IB	**A**ddress **2**. Input to store data from the input IA1 at the output O.
10+*C2*	A*C2*	IB	**A**ddress ***C2***. Input to store data from the input IA*C2* at the output O.
31	IA1	I*C1*	**I**nput **A**ddress **1**. The first input to the selector.
32	IA2	I*C1*	**I**nput **A**ddress **2**. The second input to the selector.
: 30+*C2*	IA*C2*	I*C1*	**I**nput **A**ddress *C2*. The last input to the selector.
50	O	O*C1*	**O**utput. Data output from the selector.

Function, Addressing

The input data value (A1 - IA*C2*) that is to be stored at the output is specified with the inputs A1 - A*C2*. If the input A1 is 1, the value from input IA1 is stored at the output, if A2 is 1, the value from input IA2, etc. If none of the inputs A1 - A*C2* is set to 1, the data value 0 is stored at the output O.

Loading

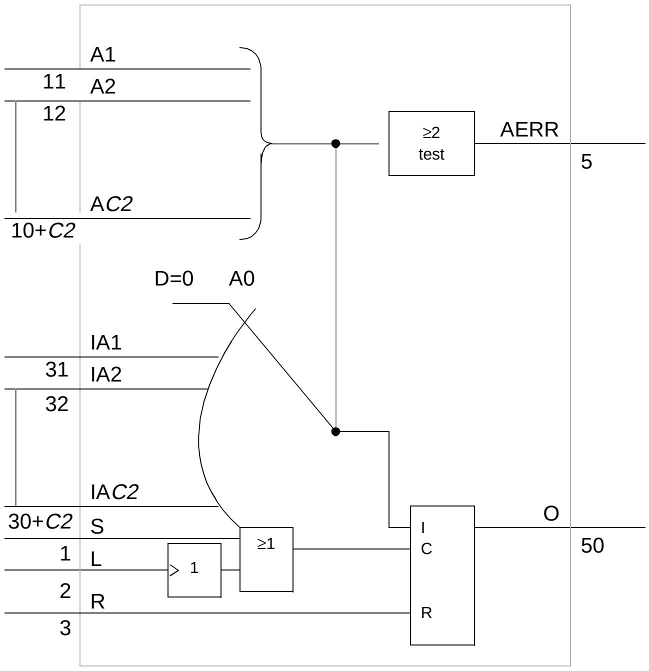
The value is loaded at the instant the input L is set to 1. If input S is set, new data is loaded each time the element is executed. When S is reset after having been set, the data loaded most recently remains. Input S overrides input L, that is, when S is set, L has no function.

Clearing

The input R clears the data output and prevents further storage of data.

Supervision

If two or more of the inputs A1 - A*C2* are set simultaneously, the output AERR is set and the value from the input which correspond to the address input set with the lowest number is stored at the output O.

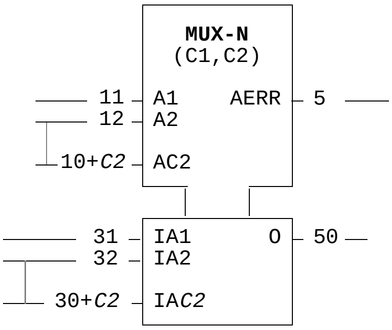


Function diagram

Multiplexer with 1 of N Address MUX-N

Summary

MUX-N (**MU**ltiple**X**er **-** with 1-of-**N** address) is used as a selector and has an optional number of inputs. The data type can be integers, real numbers, Boolean or time.



PC element MUX-N

Call MUX-N (C1, C2)

Parameter	Significance	Permissible values
C1	Data type	I, IL, R, B, T, TR
C2	Number of inputs	1 - 19

Terminal Description

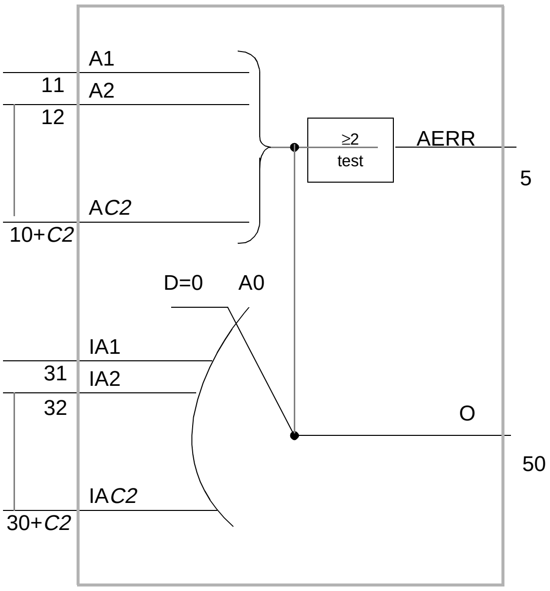
No	Name	Type	Description
5	AERR	OB	**A**ddress **ERR**or. Output which is set when two or more of the inputs A1 - A*C2* are set.
11	A1	IB	**A**ddress **1**. Input for connection of input IA1 to the output O.
12	A2	IB	**A**ddress **2**. Input for connection of input IA2 to the output O.
10+*C2*	A*C2*	IB	**A**ddress ***C2***. Input for connection of input IAC2 to the output O.
31	IA1	I*C1*	**I**nput **A**ddress **1**. The first input to the selector.
32	IA2	I*C1*	**I**nput **A**ddress **2**. The second input to the selector.
: 30+*C2*	IA*C2*	I*C1*	**I**nput **A**ddress ***C2***. The last input to the selector.
50	O	O*C1*	**O**utput. Data output from the selector.

Function, Addressing

Which one of the inputs IA1 - IA*C2* is to be connected to the output O is selected with the inputs A1 - A*C2*. If input A1 is set, input IA1 is connected to the output And if A2 is set, input IA2 is connected, etc. If none of the inputs A1 - A*C2* is set, the data value 0 is connected to the output O.

Supervision

If two or more of the inputs A1 - A*C2* are set at the same time, the output AERR is set and the input which corresponds to the address input set with the lowest number is connected to the output O.

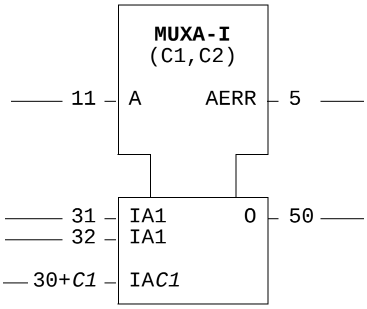


Function diagram

Multiplexer for Text MUXA-I

Summary

MUXA-I (**MU**ltiple**X**er **A**rray **-** with **I**nteger address) is used as a selector for text arrays and has an optional number of inputs. The number of characters in the text arrays is optional.



PC element MUXA-I

Call MUXA-I (C1, C2)

Parameter	Significance	Permissible values
C1	Number of inputs	1 - 19
C2	Number of characters in the arrays	1 - 80

Terminal Description

No	Name	Type	Description
5	AERR	OB	**A**ddress **ERR**or. Output which is set to one if the value at the address input A is greater than the number of inputs or negative.
11	A	II	**A**ddress. Input for address.
31	IA1	IAC2	**I**nput **A**ddress **1**. The first input in the selector.
32	IA2	IAC2	**I**nput **A**ddress **2**. The second input in the selector.
: 30+*C1*	IA*C1*	IAC2	**I**nput **A**ddress **C1**. The last input in the selector.
50	O	OAC2	**O**utput. Output from the selector.

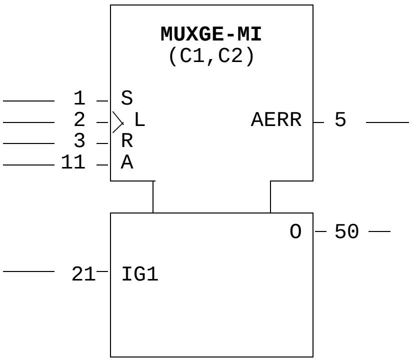
Function

The element takes addresses given at the input A and stores data from the corresponding input at output O. If the address is greater than the number of inputs or if it is negative, the output AERR is set to 1. In addition a blank text string is stored on the output. If A is set to 0, a blank text string is issued.

Multiplexer of Group Data Elements MUXGE-MI

Summary

MUXGE-MI (**MU**ltiple**X**er for **G**roup data **E**lements **-** with **M**emory and **I**nteger address) is used as a data selector with memory function**.** The element has an input for group data with an optional number of elements. The data type may be integers, real numbers, Boolean variables or time.



PC element MUXGE-MI

Call MUXGE-MI (C1, C2)

Parameter	Significance	Permissible values
C1	Data type	I, IL, R, B, T, TR
C2	Number of elements in the group vector	1 - 255

Terminal Description

No	Name	Type	Description
1	S	IB	**S**et. Input for loading of a new value each time the element is executed. When the input S is 0 the last loaded value remains at the output.
2	L	IB	**L**oad. Dynamic input for loading of addressed data.
3	R	IB	**R**eset. Input for clearing of the outputs. R overrides the inputs S and L.
5	AERR	OB	**A**ddress **ERR**or. Output which is set to 1 if the address is greater than the number of elements or is negative.
11	A	II	**A**ddress. Input for address value that indicated from which input data is to be loaded. If A=0 the output is set equal to 0.
21	IG1	IG*C1C2*	**I**nput **G**roup. Input for group data.
50	O	O*C1*	**O**utput. Data output from the selector.

Function, Addressing

At the input A is given, with an address (integer 1 - *C3*), the element of the group data the value of which is to be stored at the output. If A is zero, the data value 0 is stored.

Loading

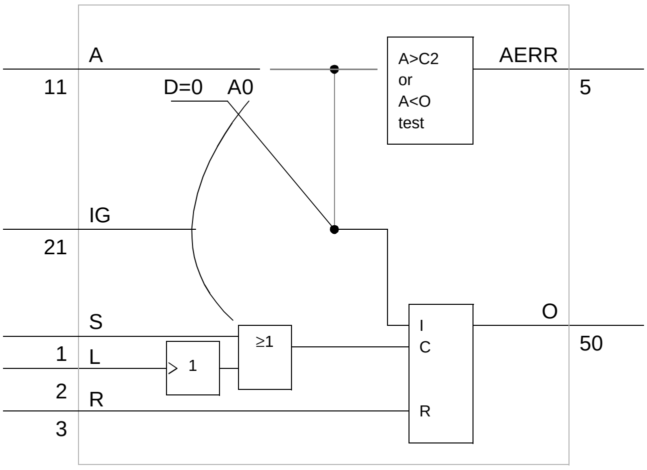
The value is loaded at the instant the input L is set to 1. If input S is set, new data is loaded each time the element is executed. When S is reset after having been set, the last data loaded remains until the element is again executed with input S, L or R set to 1. Input S overrides input L, that is, when S is set, L has no function.

Clearing

The input R clears the data output and prevents further storing of data.

Supervision

The address A is supervised and if its value is larger than the number of elements or is negative, the error signal output AERR is set. The data value 0 is then stored at the output.

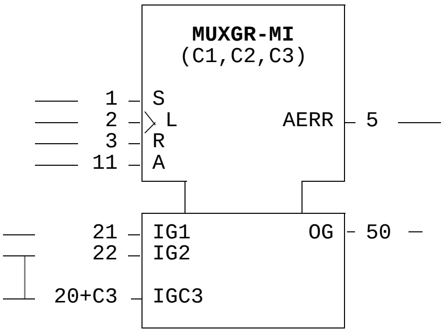


Function diagram

Multiplexer of Group Data Vectors MUXGR-MI

Summary

MUXGR-MI (**MU**ltiple**X**er for **GR**oup data vectors **-** with **M**emory and **I**nteger address) is used as a data selector with memory function. The element has an optional number of inputs. The data type may be integers, real numbers, Boolean variables or time.



PC element MUXGR-MI

Call MUXGR-MI (C1, C2, C3)

Parameter	Significance	Permissible values
C1	Data type	I, IL, R, B, T, TR
C2	Number of elements in the group vector	1 - 255
C3	Number of vectors	1 - 19

Terminal Description

No	Name	Type	Description
1	S	IB	**S**et. Input for loading of a new value each time the element is executed. When the input S is 0 the last loaded value remains at the output.
2	L	IB	**L**oad. Dynamic input for loading of addressed data.
3	R	IB	**R**eset. Input for clearing of the outputs. R overrides the inputs S and L.
5	AERR	OB	**A**ddress **ERR**or. Output which is set to 1 if the address is greater than the number of elements or is negative.
1	A	II	**A**ddress **ERR**or. Input for address value that indicates from which input data is to be loaded. If A=0, the output is set equal to 0:
21	IG1	IG*C1C2*	**I**nput **G**roup **1**. The second input of the selector.
22	IG2	IG*C1C2*	**I**nput **G**roup **2**. The second input of the selector.
: 20+*C3*	IG*C3*	IG*C1C2*	**I**nput **G**roup C3. The last input of the selector.
50	OG	OG*C1C2*	**O**utput **G**roup. Data output from the selector.

Function, Addressing

At the input A is given, with an address (integer 1 - C3), the input data value of which is to be stored at the output. If A is zero, the data value 0 is stored.

Loading

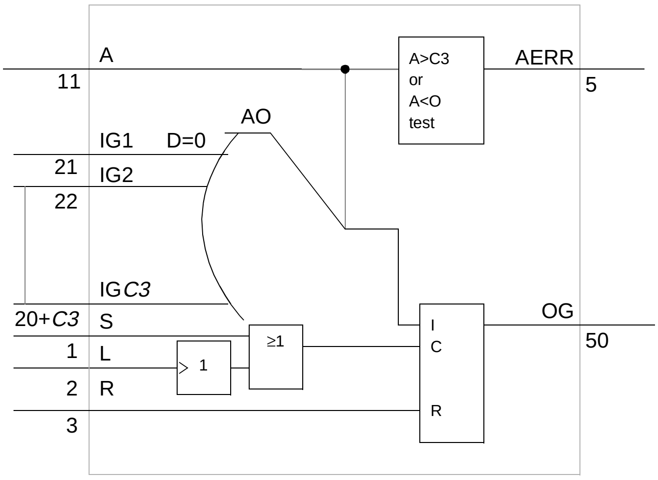
The value is loaded at the instant the input L is set to 1. If input S is set, new data is loaded each time the element is executed. When S is reset after having been set, the last data loaded remains until the element is again executed with input S, L or R set to 1. Input S overrides input L, that is, when S is set, L has no function.

Clearing

The input R clears the data output and prevents further storing of data.

Supervision

The address A is supervised and if its value is larger than the number of elements or is negative, the error signal output AERR is set. The data value 0 is then stored at the output.



Function diagram

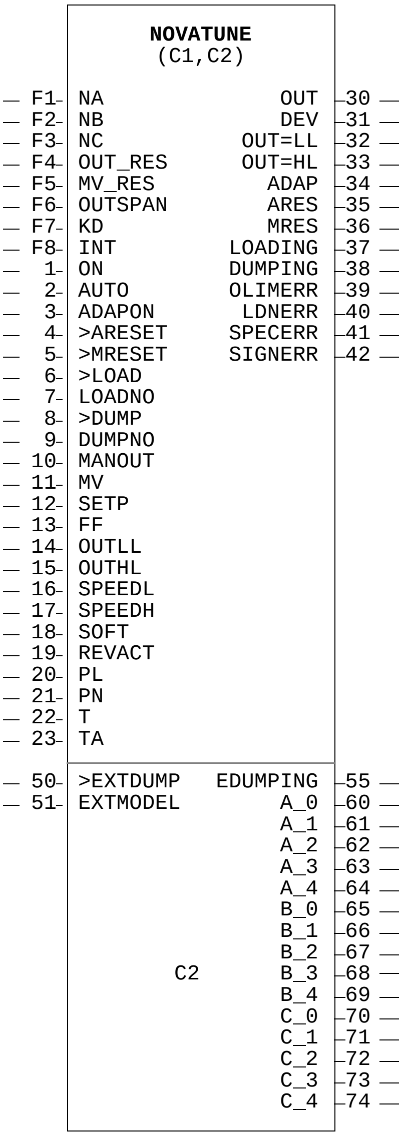
Adaptive Controller NOVATUNE

Summary

NOVATUNE is a self adjusting adaptive controller, capable of tuning its own (internal) controller parameters for optimum control.

The adaption or self-tuning takes the form of the controller generating a model of the process internally. Using this model the controller can predict what the measured value will be a certain time ahead, the prediction horizon.

The controller uses the predicted measured value to determine the new control value. Present and earlier values of control value, measured value and (where appropriate) feed forward signal are used for the prediction.



PC element NOVATUNE

Call NOVATUNE (C1, C2)

Parameter	Significance	Permissible values
C1	Number of process models that can be stored.	1 - 10
C2	Selects connection 50-74, that is, allows an external process model to be loaded and dumped and the working process model to be displayed.	0 - 1

Terminal Description

No	Name	Type	Description
F1	NA	II	Number of terms for measured value in the process model. Permitted values are 1 - 5.
F2	NB	II	Number of terms for control values in the process model. Permitted values are 1 - 5.
F3	NC	II	Number of terms for feed forward signal (if any) in the process model. Permitted values are 0 - 5.
F4	OUT_RES	IR	**OUT**put **RES**olution. Dead band of control value for adaption. Permitted values are OUT_RES>0.
F5	MV_RES	IR	**M**easured **V**alue **RES**olution. Dead band of measured value for adaption. Permitted values are MV_RES>0.
F6	OUTSPAN	IR	**OUT**put **SPAN**. Intended span of control value.
F7	KD	II	The step response of the closed loop system is expected to have a dead time of (KDz1) samples. Permitted values are 1 - 5.
F8	INT	IB	**INT**egration. INT=1: for integrating action in the controller. INT=0: for no integrating action.
1	ON	IB	ON=1: controller activated. ON=0: controller off, control value set to zero. NB: When the PC element is inserted in an executing PC program, all other inputs must be connected before ON is set to 1.
2	AUTO	IB	AUTO=1: automatic operating mode, that is, control with feedback. AUTO=0: manual operating mode, OUT=MANOUT.
3	ADAPON	IB	**ADAP**tion **ON**. ADAPON=1: adaptation is on. ADAPON=0: adaptation is off.
4	ARASET	IB	**A**daptor **RESET**. Input to initiate adaptation algorithm.
5	MRESET	IB	**M**odel **RESET**. Input for resetting the model parameters in the working process model. Stored models are not affected.
6	LOAD	IB	**LOAD** model. Input for loading a saved process model.
7	LOADNO	II	**LOAD** model **N**umber. Input to choose which process model is to be loaded when the LOAD input is activated. Note that LOADNO should always have a relevant value since an automatic LOAD takes place if b_0 receives the wrong sign on adaption.
8	DUMP	IB	**DUMP** model. Input for saving a copy of the working process model.

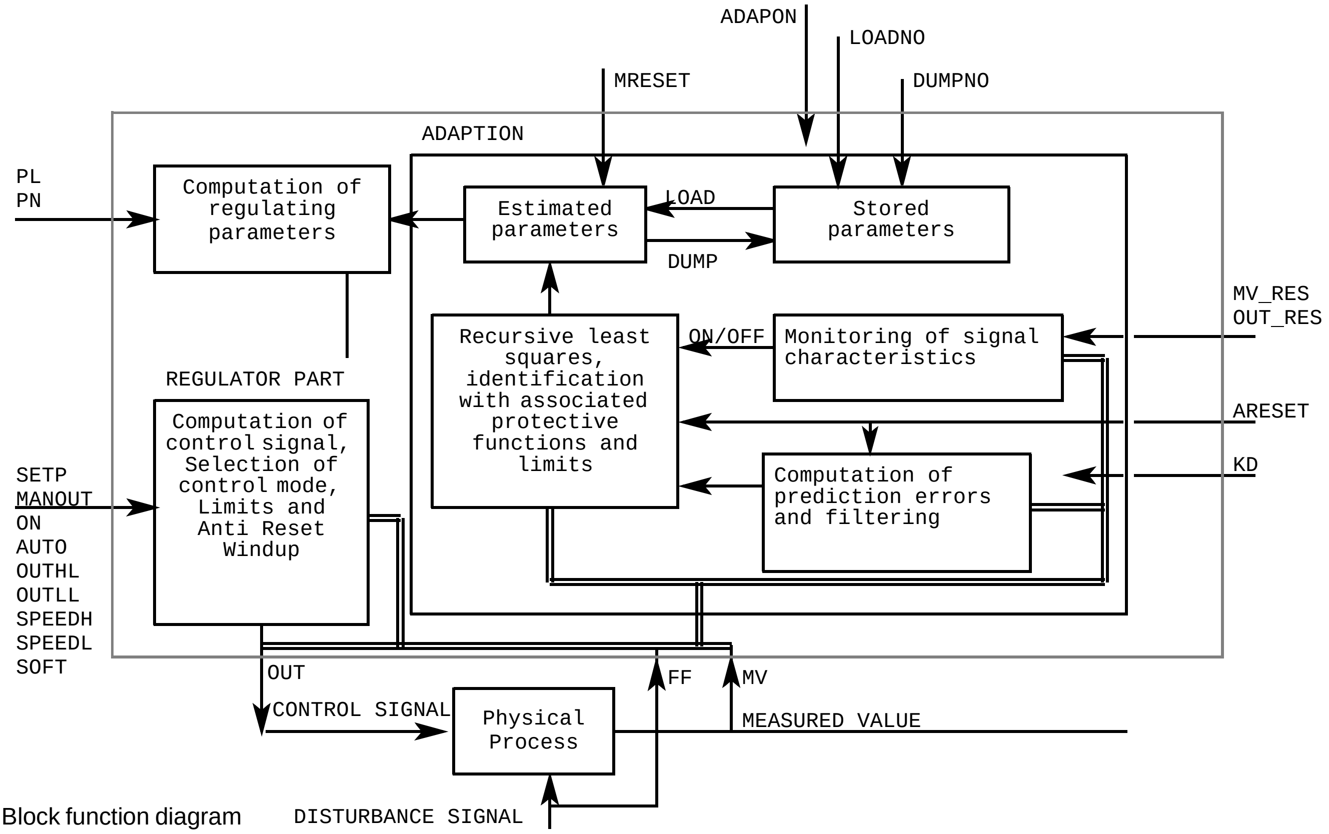
Terminal Description

No	Name	Type	Description
9	DUMPNO	II	**DUMP** model **N**umber. Input to choose where the working process model is to be saved when DUMP is activated, or where an external process model is to be saved when EXTDUMP is activated.
10	MANOUT	IR	**MAN**ual **OUT**put. Control value in manual mode.
11	MV	IR	**M**easured **V**alue. Input for actual value.
12	SETP	IR	**SETP**oint. Input for set point.
13	FF	IR	**F**eed **F**orward. Input for feed forward signal.
14	OUTLL	IR	**OUT**put **L**ow **L**imit. Lower limit of control value.
15	OUTHL	IR	**OUT**put **H**igh **L**imit. Upper limit of control value.
16	SPEEDL	IR	Greatest permitted downward step of control value in auto mode (SPEEDL≥0).
17	SPEEDH	IR	Greatest permitted upward step of control value in auto mode (SPEEDH≥0).
18	SOFT	IB	SOFT=1: results in soft start, a "window" around the initial value is opened slowly after transition to ON or AUTO. SOFT=0: gives no soft start.
19	REVACT	IB	**REV**erse **ACT**ion. REVACT=1: gives reverse control action. REVACT=0: gives direct control action.
20	PL	IR	**P**o**L**e. The value of one pole of the closed loop sampled system. Permitted values 0≤PL<1. The greater the chosen value of PL, the slower the closed loop system will be (if T x the cycle time of the execution unit is kept constant).
21	PN	II	**P**e**N**alty. "Control value penalty". Permitted values PN≥0. PN=0: gives minimum variance in the sampled control error.
22	T	II	The sampling period of the controller is T x the cycle time of the execution unit. Permitted values T≥1.
23	TA	II	The sampling period of the adaptor is TA x the cycle time of the execution unit. Permitted values TA≥1.The effective value of TA is always an integer multiple of T.
30	OUT	OR	**OUT**put. Control value.
31	DEV	OR	**DEV**iation. Output of control deviation, DEV=MV–SETP.
32	OUT=LL	OB	**OUT**put = **L**ow **L**imit. Output which is set to 1 if the control value is equal to OUTLL.
33	OUT=HL	OB	**OUT**put = **H**igh **L**imit. Output which is set to 1 if the control value is equal to OUTHL.
34	ADAP	OB	**ADAPta**tion. Output which is set to 1 when the adaptor is adapting.
35	ARES	OB	**A**daptor **RES**et. Output which is set to 1 when the adaptor is initialized.
36	MRES	OB	**M**odel **RES**et. Output which is set to 1 when the working process model is reset.
37	LOADING	OB	**LOADING** model. Output which is set to 1 when a stored process model is loaded into the adaptor.(1)
38	DUMPING	OB	**DUMPING** model. Output which is set to 1 when the working process model is being dumped from the adaptor.(1)

Terminal Description

No	Name	Type	Description
39	OLIMERR	OB	**O**utput **LIM**it **ERR**or. Output which is set to 1 when incorrect limits are chosen, that is, if OUTLL>OUTHL or if SPEEDH or SPEEDL<0.
40	LDNERR	OB	**L**oa**D N**umber or dump Number **ERR**or. Output which is set to 1 when LOADNO or DUMPNO has an illegal value, at the same time as LOAD, DUMP or EXTDUMP is activated.*
41	SPECERR	OB	**SPEC**ification **ERR**or. Output which is set to 1 when any of the inputs PL, PN, T or TA has an illegal value. Permitted values are: $0 \leq PL < 1$ $PN \geq 0$ $T \geq 1$ $TA \geq 1$
42	SIGNERR	OB	**SIGN ERR**or. Output which is set to 1 if the loaded process model has the wrong sign for b_0 relative to REVACT. At the same time the sign of b_0 is changed. The output is also set if the sign of b_0 becomes wrong during adaptation. If this happens, an automatic LOAD of the process model indicated by LOADNO takes place as well.*
50	EXTDUMP	IB	**DUMP EXT**ernal model. Input for reading in an external process model. The process model is read-in as a saved process model with the number specified by the input DUMPNO.
51	EXTMODEL	IG15R	**EXT**ernal **MODEL**. Group data input for external process model.
55	EDUMPING	OB	**E**xternal model **DUMPING**. Output which is set to 1 when an external model is being read-in.(1)
60	A_0	OR	Model parameter a_0 in the working process model.
61	A_1	OR	Model parameter a_1 in the working process model.
62	A_2	OR	Model parameter a_2 in the working process model.
63	A_3	OR	Model parameter a_3 in the working process model.
64	A_4	OR	Model parameter a_4 in the working process model.
65	B_0	OR	Model parameter b_0 in the working process model.
66	B_1	OR	Model parameter b_1 in the working process model.
67	B_2	OR	Model parameter b_2 in the working process model.
68	B_3	OR	Model parameter b_3 in the working process model.
69	B_4	OR	Model parameter b_4 in the working process model.
70	C_0	OR	Model parameter c_0 in the working process model.
71	C_1	OR	Model parameter c_1 in the working process model.
72	C_2	OR	Model parameter c_2 in the working process model.
73	C_3	OR	Model parameter c_3 in the working process model.
74	C_4	OR	Model parameter c_4 in the working process model.

(1) The outputs LOADING, DUMPING and EDUMPING remain 1 until the activity has ended or failed, but for at least one cycle of the execution unit. The outputs LDNERR and SIGNERR become active for at least one cycle at the relevant error.


ADAPON
LOADNO
MRESET
DUMPNO
ADAPTION
PL
PN
Computation of regulating parameters
Estimated parameters
LOAD
Stored parameters
DUMP
MV_RES
OUT_RES
Recursive least squares, identification with associated protective functions and limits
ON/OFF
Monitoring of signal characteristics
REGULATOR PART
ARESET
Computation of control signal, Selection of control mode, Limits and Anti Reset Windup
KD
Computation of prediction errors and filtering
SETP
MANOUT
ON
AUTO
OUTHL
OUTLL
SPEEDH
SPEEDL
SOFT
OUT
FF
MV
CONTROL SIGNAL
Physical Process
MEASURED VALUE
DISTURBANCE SIGNAL

Block function diagram

Operation, General

The adaptivity of NOVATUNE is based on the element itself creating a model of the controlled process (the object) and this model being kept up to date by the parameters following when the dynamics of the process are changed. You can say that the controller adapts by listening and learning. It knows what control values are influencing the process, since it generates them itself. It also knows the values of the feed forward disturbance and the measured values obtained, since these are input signals to the element. It learns how control values and feed forward signals affect each other.

The model describes this action. Values from a number of instants can be used to predict a future value a given time in advance. The prediction is used in two ways, on calculating new control values and on adaption.

The control value to be calculated is included in the prediction of the measured value KD samples ahead in time. It is calculated to give a suitable predicted measured value (with regard to the set point and parameter settings).

When the corresponding actual value is read, KD samples later, it can be compared with what the prediction said it would be. The difference is used to adjust the parameter values in the model.

This control technique is known as minimum variance control, since if the control value is chosen so that the predicted value is always made equal to the set point, the variance of the sampled control error is minimized. This means, among other things, that extremely good dead time compensation is included. NOVATUNE deviates from pure minimal variance control by placing a discrete time pole in PL (instead of in 0), and by being able to permit greater variance in the sampled control error in the interest of smaller control actions (controlled by the parameter PN).

With the help of the pole PL, well specified tracking of varying set points is obtained. The step response for the closed system is expected first to have a dead time of KD-1 samples. After this the control error is multiplied by PL for each new sample. This is the meaning of the pole PL.

The method that NOVATUNE uses to update the model parameters is a recursive least squares method with "forgetting factor". The forgetting factor means that the most recent signal values have the greatest effect on the model. Their effect is exponentially less and less with time. In this way the model is always adapted best to the most recent data; in other words NOVATUNE adapts itself on the basis of varying process dynamics by learning from the recent behavior of the process.

There is a major difference between this approach and adaptation by parameter schedules ("gain scheduling", referred to by some manufacturers as "adaptive gain"), where a list of gains that are to be applied in different situations is drawn up in advance. Gain scheduling gives far quicker adaption to new dynamics, but it presupposes that there is already a set of controller parameters for the new dynamics; that it is known in advance. The pure least squares method is supplemented in NOVATUNE with actions that make it more robust, among other things conditional adaptation and return to a saved model ("dump model").

Control Algorithm

The control value is calculated on the basis of the following expression when integrating action is included (INT=1).

$u_t = u_{t-1} + b_0 / (b_0^2 + \rho)\ [(1-p)(r_t-y_t)$ -

$- b_1 (u_{t-1} - u_{t-2})- ... -b_{NB-1} (u_{t-NB+1} - u_{t-NB})$ -

$- a_0 (y_t -y_{t-1}) -... -a_{NA-1} (y_{t-NA+1} - y_{tzNA})$ -

$- c_0 (v_t -v_{t-1}) -... -c_{NC-1} (v_{t-NC+1} -v_{t-NC})]$

The following expression applies without integrating action (INT=0).

$u_t = [\rho / (b_0^2 + \rho)]\ u_{t-1} + b_0 / (b_0^2 + \rho)[(1-p)r_t$ -

$-b_1\ u_{t-1} -... -b_{NB-1}\ u_{t-NB+1}$ -

$- a_0\ y_t -... -a_{NA-1}\ y_{t-NA+1}$ -

$- c_0\ v_t -... -c_{NC-1}\ v_{t-NC+1}]$

u is the output signal (control value) of the controller=OUT

y is the process variable (measured value)=MV

v is the feedforward signal=FF

r is the reference value (setpoint)=SETP

The number of parameters in the model is:

for the measured value=NA

for the control value=NB

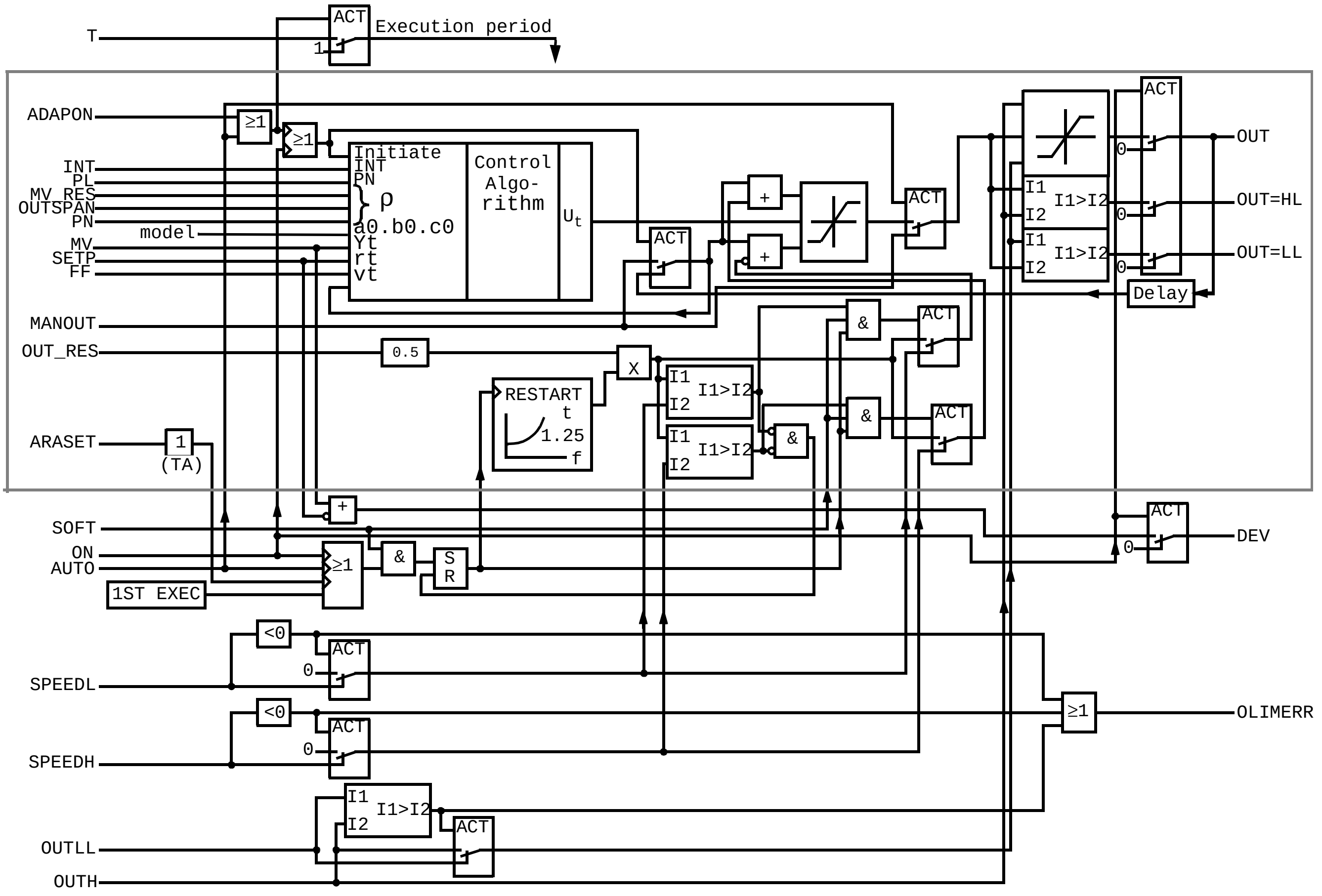
for the feedforward signal=NC

p is the pole = PL

The weight (the penalty) ρ [rho] is determined by the parameter "penalty" = PN

the weight $\rho = \rho_0 x2^{\mathbf{PN}}$

the basic value ρ_0=4xMV_RES$^{\mathbf{2}}$/(OUTSPAN)$^{\mathbf{2}}$



Block diagram of controller section

Operating Modes

Like most conventional controllers, NOVATUNE can also be operated manually (AUTO=0). This means that the input MANOUT directly feeds to the output OUT. Normally, manual mode must not be used outside NOVATUNE.

If you do this, it is important to tell the NOVATUNE element that the process is being run manually (AUTO=0) and to give the value of the manual output signal (MANOUT). If the element cannot obtain information about relevant manual control values, the adaptation must be switched off (ADAPON=0). If you do not, the controller adapts to the wrong information.

It is also essential for the element to receive the manual control value so that a bumpless transition can be made back to automatic mode. Generally, adaptation should not take place during manual operation except possibly during commissioning, in which case the output signal must be chosen so that the process is excited.

If the input T is greater than 1, the manual mode is dependent on the input ADAPON. If adaptation is switched off (ADAPON=0) the output out is updated to MANOUT every time the element executes. On the other hand, if adaptation is active (ADAPON=1) OUT is updated only at the normal execution period (T x cycle time of execution unit). This is because, if the signal varies between sampling instants, the adaptation gives a model that deviates from the desired model.

Adaption

In some cases there is a need to stop further updating of the internal model of the controller. The ADAPON input provides a means of controlling this with logical conditions. There are also a number of safety and robustness aspects concerning adaption. Always using adaptation and always using the "forgetting factor" gives a controller that is oversensitive to certain types of disturbance. The PC element includes a number of features to increase its robustness.

Adaptation is conditional; this means that it is switched off when there is no excitation, that is, when there are no signal variations for the controller to learn from. The same applies if the control error is already as small as it is required to be, in which case there is no reason for further adjustment. The limit for not regarding the measured value as excited or for regarding the controller error as sufficiently small =MV_RES. The limit for not regarding the control value as exciting =OUT_RES.

The "forgetting factor" has been mentioned once or twice. At its basic value, data is "forgotten" with a time constant of 50 adaptions. An adaptation can be made less frequently than a new control value is calculated (with regard to processor load, specified with TA), and this must be taken into account if an idea of the time over which the model is adapted is required. Similarly, it is essential to take account of whether the limits MV_RES or OUT_RES have switched off adaptation. If so, data has not been "forgotten" under that period either. It may also happen that the model is considered so uncertain that all data shall be used for updating. In this case the "forgetting factor" will not be used until the model has been determined with greater certainty. There are also features to prevent sudden disturbances changing the model too rapidly.

Reverse/Direct Control Action

For a process in which a change in control value results in a measured value response in the same direction, a controller with reverse action is required (REVACT=1). This means that b_0 must be positive. The opposite, direct action (REVACT=0), corresponds to negative b_0. This is a fundamental property of the controller. If an attempt is made to start the controller with the wrong action, the control value will rapidly go to limit. There would not be enough time for the controller to learn that it was wrong and once the control value is steady, there is no adaption because of lack of excitation. The PC element prevents spontaneous change of sign in b_0. If the adaptation leads to such a change, it is never carried into effect. Instead, the "DUMP" model is loaded (an automatic "LOAD" is performed). If the sign is incorrect because of a change of REVACT or loading of a saved model with incorrect sign, the sign of b_0 is changed. It is therefore essential to specify whether reverse or direct action (positive or negative b_0) is required. The errors described above give indication in SIGNERR.

Initializing

Activation of ARESET is a signal to the element not to rely on the current working model but instead to adapt as quickly as possible to a situation which is to be regarded as new. Activation of MRESET resets the model to b_0=1 for reverse action (REVACT=1) of b_0=z1 for direct action (REVACT=0), and the remaining model parameters=0.

Execution of MRESET, DUMP, EXTDUMP, LOAD and sign change on change of REVACT all take place in the first part of the adaptor. If the adaptor is occupied with later stages of execution when the request comes, it may take up to the time given by TA before the functions are executed, even if the controller section performs one or more interventions in between. When DIBM is performed on the control model that covers NOVATUNE, the element will be initialized. This includes the loading of the model specified by LOADNO.

Load and Dump

The intention is not that the controller should see the process as completely foreign if the control system has to be restarted, or if the process is restarted from an operating case that should be known. A model is therefore saved by requesting DUMP.
It is retrieved on startup or when LOAD is requested.

In NOVATUNE, "Gain Scheduling" can be achieved by using several dump models and adapting them to different operating cases. The LOAD input is used to reload the required model (LOADNO) and directly replaces the model that had been obtained at the time by adaptation. After this adaptation continues in the usual way from the reloaded model.

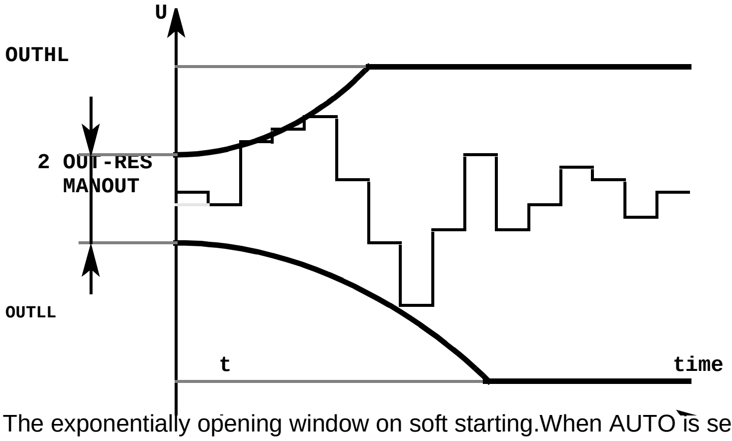
Signal Limitations

When the adaptive controller updates its model of the process, it presupposes that the output signal leaving the controller is the one that influences the process. This means that the output signal of the controller must not be limited at any later stage.

With NOVATUNE, it is possible to set the lower (OUTLL) and upper (OUTHL) limits of the output signal. In addition, limits can be set for decrease (SPEEDL) and increase (SPEEDH) of the output signal per controller intervention in auto mode.
These are specified as an absolute value of maximum permitted change at one sampling instant. Both are thus positive or zero.

On commissioning it is often appropriate to have a narrower limitation of the output signal until the controller has adapted.
This can be done automatically by means of a built in soft start function (SOFT). This means that the permitted changes of the output signal increase exponentially from the start value OUTRES to SPEEDL or SPEEDH.

Alternatively, the narrower limits are set with the inputs OUTLL and OUTHL, which gives manual control of when to increase the permitted range.



The exponentially opening window on soft starting.When AUTO is set to 1 at time t, the window first opens around the latest manual output signal (MANOUT).

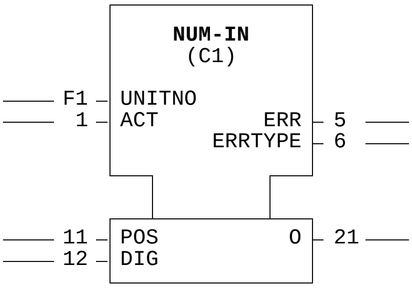
Tuning

For information on tuning the NOVATUNE PC element, refer to the manual ”Installation, Commissioning and Service”.

Numerical Input from Thumbwheel Switch NUM-IN

Summary

NUM-IN (**NUM**eric-**IN**put) is used for the acquisition of values from thumbwheel unit DSHM 140.



PC element NUM-IN

Call NUM-IN (C1)

Parameter	Significance	Permissible values
C1	Data type of value to be output	I, IL

Terminal Description

No	Name	Type	Description
F1	UNITNO	FI	**UNIT N**umber. Address for MasterView 100 unit. The hundreds and thousands indicate the number of the MasterView 100 bus, the units and tens indicate the number of the node.
1	ACT	IB	**ACT**ivate. Input to activate reading from the thumbwheel switch.
5	ERR	OB	**ERR**or. Output for indication of error.
6	ERRTYPE	OI	**ERR**or **TYPE**. Output to determine type of error.
11	POS	II	**POS**ition. Input for position of first digit to be read from the thumbwheel switch.
12	DIG	II	**DIG**its. The number digits to be read in.
21	OC1	OV*C1*	**O**utput. Output for value entered.

Function

NUM-IN is used for the entry of numeric values from thumbwheel unit DSHM 140. Integer (I) and long integers (IL) can be entered with the element. The type is specified with call parameter C1. Each NUM-IN element can control a part or all of a DSHM 140. The position on the thumbwheel switch where the value is to be read is controlled from the inputs POS and DIG.

POS indicates which digit is the first to be controlled by this element. The characters are numbered from the left from 1 to 6. Character 1 is +/- and characters 2-6 are the numerals 0 - 9.

Activation of Reading

If the ACT input is set to 1, a new value is read from the thumbwheel switch each time the element is executed. If ACT=0, the original value remains at output O.

Error Supervision

When the element indicates an error, the error output ERR is set to 1, and the type of error is indicated at the same time at the ERRTYPE output. The element can indicate the following error types:

0=The element has not detected an error (ERR=0)

1=Controller board to specified MasterView 100 bus missing

2=Controller board to specified MasterView 100 bus faulty

3=The addressed MasterView 100 unit is missing or faulty

11=Illegal value at POS input

12=Illegal value at DIG input

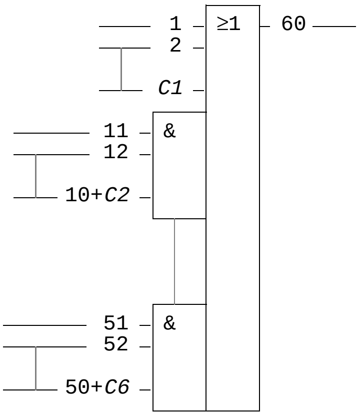
15=Illegal value at function parameter UNITNO

99=The value at the output is limited to the maximum positive/ negative value that can be presented (only occurs when C1=I).

Or Gate with And Inputs OR-A

Summary

OR-A (**OR -** with **A**nd gates on the inputs) is used to form general combined expressions with Boolean variables.



PC element OR-A

Call OR-A (C1, C2, C3,... Cn)

Parameter	Significance	Permissible values
C1	Number of inputs in the OR part of the element	0 - 9
C2	Number of inputs in the first AND gate of the element	0 - 9
C3	Number of inputs in the second AND gate of the element	0 - 9
: C*n*[1]	Number of inputs in the last AND gate of the element	0 - 9

[1] $1 \le n \le 6$
$1 \le C1+C2+...+Cn \le 40$

Terminal Description

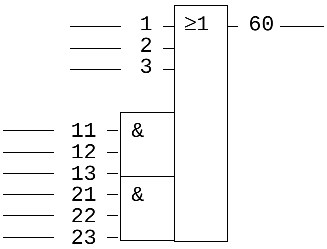
No	Name	Type	Description
1	—	IB	Input direct to the OR gate in the element.
2	—	IB	Input direct to the OR gate in the element.
: *C1*	—	IB	Input direct to the OR gate in the element.
11	—	IB	Input to the first AND gate in the element.
12	—	IB	Input to the first AND gate in the element.
10+*C2*	—	IB	Input to the first AND gate in the element.
21	—	IB	Input to the second AND gate in the element.
22	—	IB	Input to the second AND gate in the element.

Terminal Description

No	Name	Type	Description
: 20+*C3*	—	IB	Input to the second AND gate in the element.
10x(*n*-1)+1	—	IB	Input to the last AND gate in the element.
10x(*n*-1)+2	—	IB	Input to the last AND gate in the element.
: 10x(*n*-1)+C*n*	—	IB	Input to the last AND gate in the element.
60	—	OB	Output.

Function

More rapid execution than if separate elements are used can be obtained by using assembled elements such as OR-A. The inputs need then only be tested until the value of the output can be determined. If input 1 is 1, the output can be set irrespective of the other inputs. See the figure and the table below.



PC element OR-A (3, 3, 3)

Table OR-A (3, 3, 3)

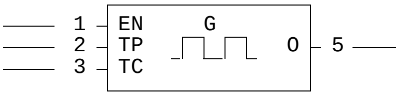
1	2	3	11	12	13	21	22	23	60
0	0	0	0	0	0	1	1	1	1
0	0	0	0	x	x	0	x	x	0
0	0	0	0	x	x	1	0	x	0
0	0	0	0	x	x	1	1	0	0
0	0	0	1	0	x	0	x	x	0
0	0	0	1	0	x	1	0	x	0
0	0	0	1	0	x	1	1	0	0
0	0	0	1	1	0	0	x	x	0
0	0	0	1	1	0	x	0	x	0
0	0	0	1	1	0	x	x	0	0
0	0	0	1	1	1	x	x	x	1
0	0	1	x	x	x	x	x	x	1
0	1	x	x	x	x	x	x	x	1
1	x	x	x	x	x	x	x	x	1

x indicates that the input has no effect on the value of the output or the execution time.

Oscillator OSC-B

Summary

Oscillator OSC-B (**OSC**illator **- B**oolean variables) with variable frequency and pulse time is used when pulse trains with periods from 2 x cycle time up to 24 hours are needed.



PC element OSC-B

Call OSC-B

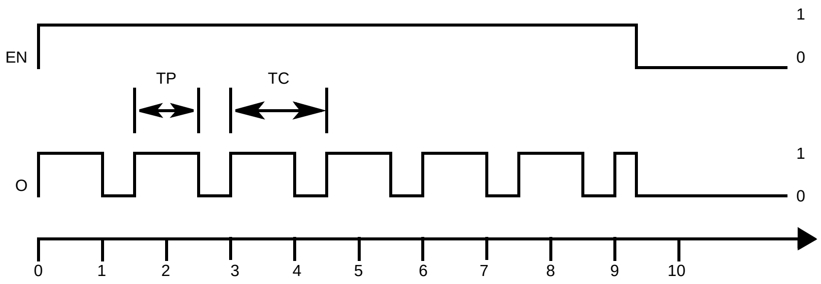
Terminal Description

No	Name	Type	Description
1	EN	IB	**EN**able. Input for start of oscillator.
2	TP	IT	**T**ime **P**ulse. Input for setting of pulse time.
3	TC	IT	**T**ime **C**ycle. Input for setting of period. The period must be greater than the pulse duration.
5	O	OB	**O**utput. Oscillator output.

Function

The oscillator starts when the input EN is set (to 1). The figure below shows an example of an oscillator time diagram.

TC and TP are rounded up to a multiple of the actual cycle time.



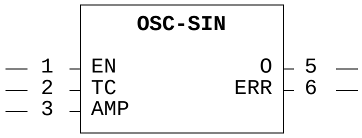
Example of time diagram with TP=1 s and TC=1.5 s

Oscillator for Sin Wave

OSC-SIN

Summary

OSC-SIN (**OSC**illator **- SIN**e wave) is used to generate sinus signals. The period and amplitude are controlled by inputs.



PC element OSC-SIN

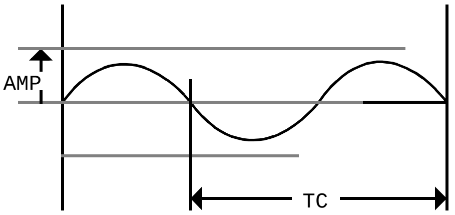
Call OSC-SIN

Terminal Description

No	Name	Type	Description
1	EN	IB	**EN**able. Input for activation of generation of sine wave.
2	TC	ITR	**T**ime **C**ycle. Input for period.
3	AMP	IR	**AMP**litude. Input for amplitude.
5	O	OR	**O**utput. Output for sine wave.
6	ERR	OB	Output which is set (to 1) if the period TC selected is too short.

Function

OSC-SIN generates a sinus wave with period TC and amplitude AMP, if the input EN=1, as shown in the figure below. The sine wave is created from a table with 256 values per period. If EN is reset (to 0) the output O is cleared. When EN is set to 1 after having been 0, the sine wave begins from t=0.



Wave form generated

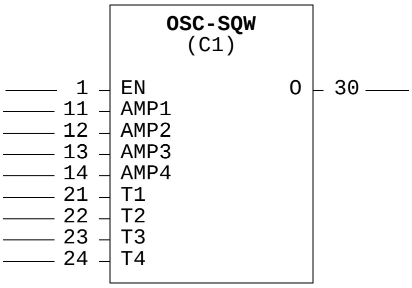
Supervision

If the element is executed with a period time TC which is less than 20 times the cycle time, the output ERR is set to 1. If the cycle time is less than the period TC, the output O is cleared.

Oscillator for Square Wave OSC-SQW

Summary

OSC-SQW (**OSC**illator **- SQ**uare **W**ave) is used to generate a square wave with up to four levels. Each level is determined by an input and for each level a time during which it is to be maintained is specified.



PC element OSC-SQW

Call OSC-SQW (C1)

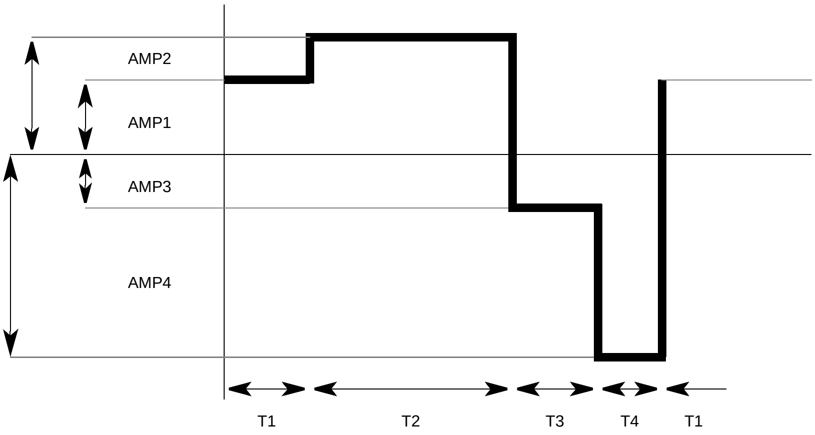
Parameter	Significance	Permissible values
C1	Specifies number of levels in the square wave	2 - 4

Terminal Description

No	Name	Type	Description
1	EN	IB	**EN**able. Input for activation of generation of square wave.
11	AMP1	IR	**AMP**litude **1**. Input for amplitude during the time T1.
12	AMP2	IR	**AMP**litude **2**. Input for amplitude during the time T2.
13	AMP3	IR	**AMP**litude **3**. Input for amplitude during the time T3.
14	AMP4	IR	**AMP**litude **4**. Input for amplitude during the time T4.
21	T1	IT	**T**ime **1**. Input for time during which AMP1 is to be available at the output.
22	T2	IT	**T**ime **2**. Input for time during which AMP2 is to be available at the output.
23	T3	IT	**T**ime **3**. Input for time during which AMP3 is to be available at the output.
24	T4	IT	**T**ime **4**. Input for time during which AMP4 is to be available at the output.
30	O	OR	**O**utput. Output for generated square wave.

Function

If input EN=1, OSC-SQW generates a square wave with up to four different amplitudes in accordance with the figure below. If EN is reset, the output O is also cleared. When EN is set to 1, after having been 0, the wave form starts from the beginning.

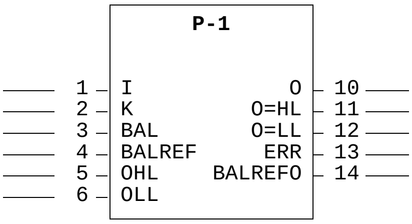


Wave form generated

Proportional Function P-1

Summary

P-1 (**P**roportional Function) is used as an amplifier in feedback systems. The output signal can be limited with limit values specified at special inputs. The balancing function permits the output signal to follow an external reference.



PC element P-1

Call P-1

Terminal Description

No	Name	Type	Description
1	I	IR	**I**nput. Input signal.
2	K	IR	Input for setting gain.
3	BAL	IB	**BAL**ance. Input for activation of following.
4	BALREF	IR	**BAL**ance **REF**erence. Input for reference value when following.
5	OHL	IR	**O**utput **H**igh **L**imit. Input for upper limit value.
6	OLL	IR	**O**utput **L**ow **L**imit. Input for lower limit value.
10	O	OR	**O**utput. Output signal.
11	O=HL	OB	**O**utput=**H**igh **L**imit. Output which is set to 1 if the output is limited to the upper limit value.
12	O=LL	OB	**O**utput=**L**ow **L**imit. Output which is set to 1 if the output is limited to the lower limit value.
13	ERR	OB	**ERR**or. Output which is set to 1 if OHL is less than OLL.
14	BALREFO	OR	**BAL**ance **REF**erence **O**utput. Information output for O/K.

Function

The P-function can be written in the time plane as O(t)=K I(t). Its main property when controlling is that the output signal is directly proportional to the input signal. The transfer function for a P-function is: G(S)=K

Following

If BAL is set to 1, the amplifier goes into following and the output O is set to the value of the input BALREF. If the value at BALREF exceeds the output signal limits, the output is set to the limit value concerned.

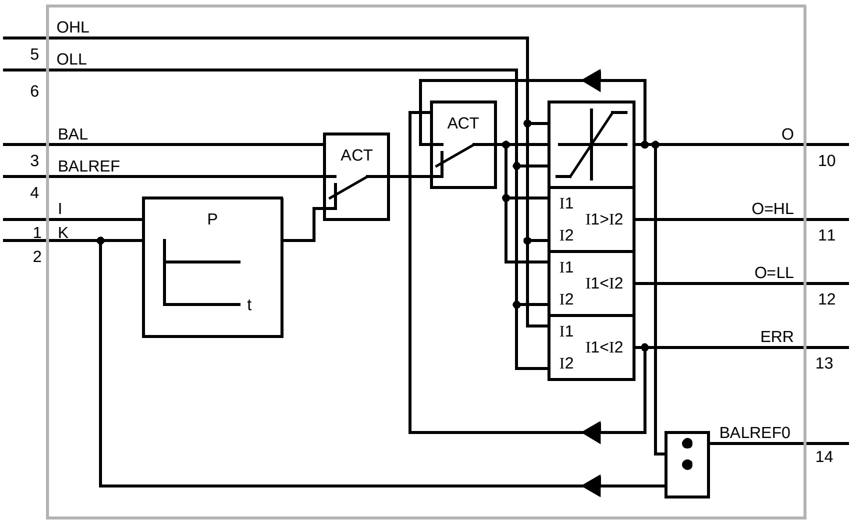
Limitation Function

The limitation function limits the output signal to the limit values at the inputs OHL for upper limit value and OLL for the lower limit value. If the actual value exceeds the upper limit, the output O=HL is set to 1 and if it falls below the lower limit, the output O=LL is set to 1.

The element checks that the upper limit OHL is greater than the lower limit OLL. If not, the output ERR is set to 1. While the error status persists, the outputs O=HL, O=LL and O retain the values they had in the sample before that in which the error occurred.

Information Output BALREFO

The output BALREFO gives the result of the calculation O(t)/K where O(t) is the output signal after the limitation. The output specifies the input signal which corresponds to the output signal concerned.

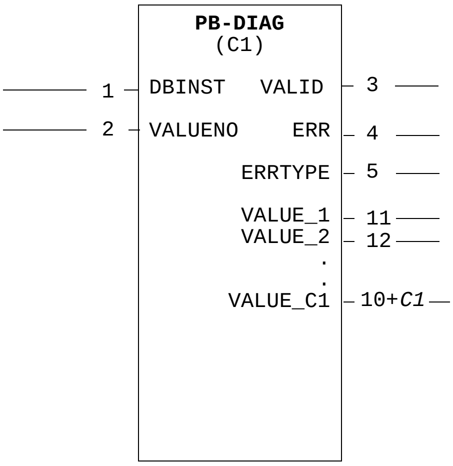


Function diagram

Receive Diagnostic Data from PROFIBUS-DP Slave PB-DIAG

Summary

PB-DIAG (**P**ROFI**B**US - **Diag**nostic) is used for access of diagnostic data from a PROFIBUS-DP slave. It uses the PROFIBUS-DP Slave DB element to get address information.



PC element PB-DIAG

Call PB-DIAG(C1)

Parameter	Significance	Permissible values
C1	Number of values	1 - 32

Terminal Description

No	Name	Type	Description
1	DBINST	II	**D**ata **B**ase **INST**ance. Input defining the instance number of the corresponding PROFIBUS-DP slave DB element. Alternatively, the name of the DB element
2	VALUENO	II	**VALUE N**umber 1 - 245. Identifies the data to be read. See **Function** below
3	VALID	OB	Data **VALID** flag.
4	ERR	OB	**ERR**or flag. See **Error handling** below
5	ERRTYPE	OI	**ERR**or **TYPE.** See **Error handling** below
11	VALUE_1	OB	Data.
... 10+C1	VALUE_C1	OB	

Function

PB-DIAG is used to access diagnostics data from a PROFIBUS-DP slave. The address of the PROFIBUS-DP slave is found in the corresponding DB element which instance reference is given by the terminal DBINST.

The element reads a diagnostic data value from the diagnostic data message received from the slave. The terminal VALUENO gives the position number of the value in the message. Each PROFIBUS-DP slave sends or receives a number of process values or attributes (of different or equal data types) packed together in a message called a PDU (Payload Data Unit, see PROFIBUS User’s Manual). The slave device manufacturer describes the contents of the PDU in the GSD file (Geräte Stamm Daten). Diagnostic data corresponding to VALUENO = 245 contain information about the slave device connection state and is generated locally in Advant Controller 450 by the PROFIBUS communication interface module.

The element puts read data on the terminals VALUE_1 - VALUE_32 and current data status on the terminal VALID. VALID is set TRUE if data are valid. The number of outputs depends on the parameter C1 (1 - 32).

Configuration

Slave address information is fetched from the DB element PROFIBUS-DP Slave with the given instance reference

Error handling

The following error codes could be reported on the ERRTYPE terminal:

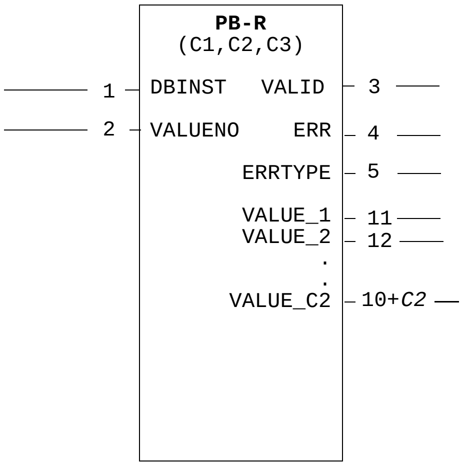
0 OK, no error
1 DB element according to DBINST is not found
2 DB element not properly initiated. Error in data.
3 DB element error marked.
4 Bus Error. The PROFIBUS interface module is not reachable.
5 The DB element is marked ‘Not Implemented’
6 The DB element is marked ‘Not In Service’.

The terminal ERR is set to 1 and VALID is set to FALSE if ERRTYPE <> 0.

Receive Data from PROFIBUS-DP Slave PB-R

Summary

PB-R (**P**ROFI**B**US - **R**eceive) is used for access of input data from a PROFIBUS-DP slave. It uses the PROFIBUS-DP Slave DB element to get address information.



PC element PB-R

Call PB-R(C1, C2,C3)

Parameter	Significance	Permissible values
C1	Data type	B,I,IL,R
C2	Number of values	1 - 32
C3	Read output values from a slave	0 - 1

Terminal Description

No	Name	Type	Description
1	DBINST	II	**D**ata **B**ase **INST**ance. Input defining the instance number of the corresponding PROFIBUS-DP slave DB element. Alternatively, the name of the DB element
2	VALUENO	II	**VALUE N**umber 1 - 244. Identifies the data to be read. See **Function** below
3	VALID	OB	Data **VALID** flag. TRUE when valid data on terminals VALUE_1 - VALUE_C2
4	ERR	OB	**ERR**or flag. See **Error handling** below
5	ERRTYPE	OI	**ERR**or **TYPE.** See **Error handling** below
11	VALUE_1	O*C1*	Data. Type depends on C1
... 10+C2	VALUE_C2	O*C1*	

Function

PB-R is used to access input data from a PROFIBUS-DP slave. The address of the PROFIBUS-DP slave is found in the corresponding DB element which instance reference is given by the terminal DBINST.

The element reads a consecutive sequence of data values (of the same type) from the data message received from the slave. The terminal VALUENO gives the position number of the first value in the message. Each PROFIBUS-DP slave sends or receives a number of process values or attributes (of different or equal data types) packed together in a message called a PDU (Payload Data Unit, see PROFIBUS User's Manual). The slave device manufacturer describes the contents of the PDU in the GSD file (Geräte Stamm Daten).

The element puts read data on the terminals VALUE_1 - VALUE_32 and current data status on the terminal VALID. VALID is set TRUE if received data is valid. The number of data read, depends on the parameters C1 and C2. C1 is the data type (B, I, IL, R) and C2 is the number of outputs (1 - 32). If C1 = B, only one data value is read and the bits (0 - C2-1) in the data are written as booleans to the C2 outputs. If C1 = (I, IL R), C2 data values are read and written.
If C3 = 1, values from the output buffer is read. This is meaningful for a PROFIBUS-DP slave which is configured with READONLY terminal set to 1 in the PBS data base element. In this case another PROFIBUS-DP master updates this slave. C3=0 means reading from the input buffer which is the default.

Configuration

Slave address information is fetched from the data base element PROFIBUS-DP slave with the given instance reference.

Error handling

The following error codes could be reported on the ERRTYPE terminal:

0 OK, no error
1 DB element according to DBINST is not found
2 DB element not properly initiated. Error in data.
3 DB element error marked.
4 Bus Error. The PROFIBUS interface module is not reachable.
5 The DB element is marked ‘Not Implemented’
6 The DB element is marked ‘Not In Service’.

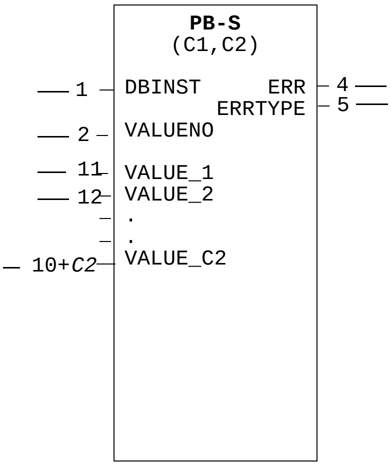
The terminal ERR is set to 1 and VALID is set to FALSE if ERRTYPE <> 0.

Send Data to PROFIBUS Slave

PB-S

Summary

PB-S (PROFIBUS - Send) is used for sending data to PROFIBUS slave. It uses the PROFIBUS-DP Slave DB element to get address information.



PC element PB-S

Call PB-S(C1, C2)

Parameter	Significance	Permissible values
C1	Data type	B,I,IL,R
C2	Number of values	1 - 32

Terminal Description

No	Name	Type	Description
1	DBINST	II	**D**ata **B**ase **INST**ance. Input defining the instance number of the corresponding PROFIBUS-DP slave DB element. Alternatively, the name of the DB element
2	VALUENO	II	**VALUE N**umber 1 - 244. Identifies the data to be written. See **Function** below
11	VALUE_1	I*C1*	Data. Type depends on C1
... 10+C2	VALUE_C2	I*C1*	
4	ERR	OB	**ERR**or flag. See **Error handling** below
5	ERRTYPE	OI	**ERR**or **TYPE.** See **Error handling** below

Function

PB-S is used to output data to a PROFIBUS-DP slave. The address of the PROFIBUS-DP slave is found in the corresponding DB element which instance reference is given by the terminal DBINST.

The element transfers data from the terminals VALUE_1 - VALUE_32 to a consecutive sequence of data values in the data message sent to the slave. The terminal VALUENO gives the position number of the first value in the message. Each PROFIBUS-DP slave sends or receives a number of process values or attributes (of different or equal data types) packed together in a message called a PDU (Payload Data Unit, see PROFIBUS User’s Manual). The slave device manufacturer describes the contents of the PDU in the GSD file (Geräte Stamm Daten).

The number of terminals read, depends on the parameter C2 and the handling depends on C1. C1 is the data type (B, I, IL, R) and C2 is the number of terminals (1 - 32). If C1 = B, the C2 inputs are transferred to the bits (0 - C2-1) in one data value. If C1 = (I, IL, R), C2 inputs are transferred to C2 data values

Configuration

Slave address information is fetched from the DB element PROFIBUS-DP Slave with the given instance reference.

WARNING!

More than one PB-S PC element can update the same slave data value. There is no checking that the combination of VALUENO and C2 gives non-overlapping positions in the PDU.

Error handling

The following error codes could be reported on the ERRTYPE terminal:

0 OK, no error
1 DB element according to DBINST is not found
2 DB element not properly initiated. Error in data.
3 DB element error marked.
4 Bus Error. The PROFIBUS interface module is not reachable.
5 The DB element is marked ‘Not Implemented’
6 The DB element is marked ‘Not In Service’.

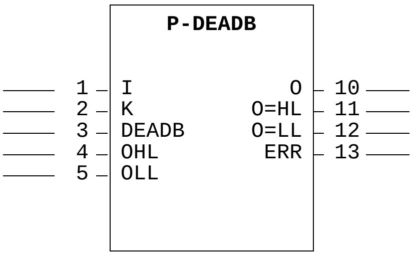
The terminal ERR is set to 1 if ERRTYPE <> 0.

Low Amplitude Filter P-DEADB

Summary

P-DEADB is used as a filter for signal changes less than a specified dead band. The output signal can be limited with limits specified at special inputs.

Call P-DEADB



PC element P-DEADB

Terminal Description

No	Name	Type	Description
1	I	IR	**I**nput. Input signal.
2	K	IR	Input for setting dead band gain.
3	DEADB	IR	**DEADB**and. Input for setting of the size of the dead band.
4	OHL	IR	**O**utput **H**igh **L**imit. Input for upper limit value.
5	OLL	IR	**O**utput **L**ow **L**imit. Input for lower limit value.
10	O	OR	**O**utput. Output signal.
11	O=HL	OB	**O**utput=**H**igh **L**imit. Output which is set to 1 if the output is limited to the upper limit value.
12	O=LL	OB	**O**utput=**L**ow **L**imit. Output which is set to 1 if the output is limited to the lower limit value.

Function

The input signal change related to the output signal is calculated as follows

ABS(I(t)-O(t-TS)).

The input signal change is compared with the dead band, DEADB, which is given a value with the same unit as the input signal. If the input signal change is greater than or equal to DEADB, the input signal is not processed, that is,

O(t)=I(t)

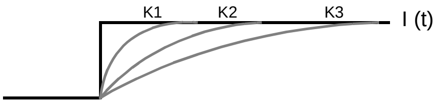
If the input signal change is less than the dead band, the signal is amplified in accordance with

O(t)=O(tzTS)+K x (I(t)zO(tzTS)),

that is, an output signal change which is dependent on the difference between the input signal and the output signal of the previous sample. This function corresponds to a first order filter.

Dead band Gain

The dead band gain, K, is given as a factor in the numerical range 0<K<1. Gain values greater than 1 are limited to 1 which means that O(t)=I(t), that is, the output signal follows the input signal also in the dead band. Gain values less than or equal to 0 are limited to 0 which means that O(t)=O(tzTS), that is, the output signal is not changed if the input signal change is within the dead band.

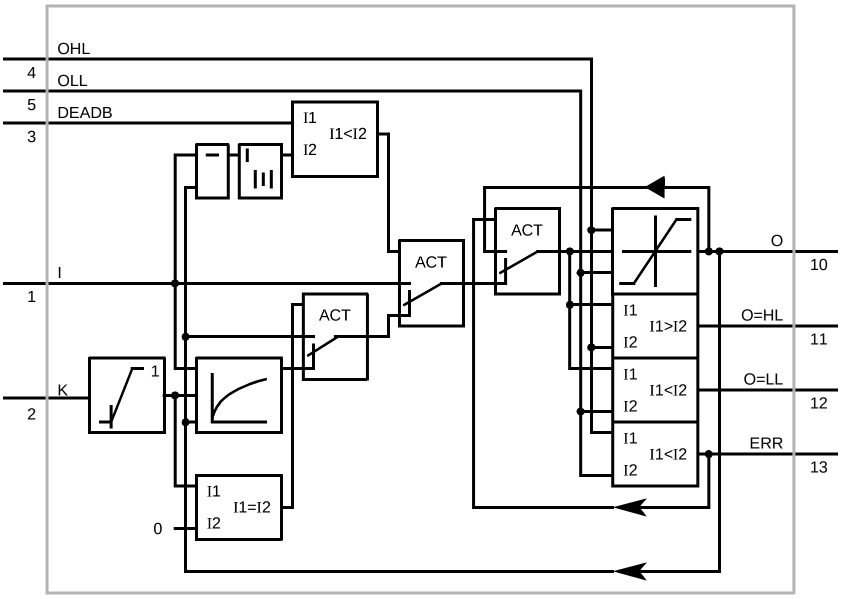


The output signal O(t) with signal change less than the deadband

Limitation Function

The limitation function limits the output signal to the limit values at the inputs OHL for upper limit value and OLL for the lower limit value. If the actual value exceeds the upper limit value, the output O=HL is set to 1 and if it falls below the lower limit value, the output O=LL is set to 1.

The element checks that the upper limit value OHL is greater than the lower limit value OLL. If not, the output ERR is set to 1. While the error status persists, the outputs O=HL, O=LL and O retain the values they had in the sample before that in which the error occurred.

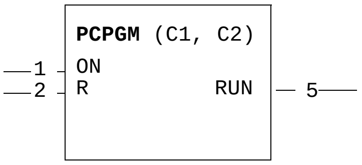


Function diagram

Program Header PCPGM

Summary

The program header PCPGM is used for the structuring and execution control of the application program of a PC system.



PC element PCPGM

Call PCPGM (C1, C2)

C1	Cyclicity in ms	Permissible values are determined by the possible cycle times of the relevant PC system. These times are specified in the manual for the system concerned.
C2	Place in the cycle time table	Permissible values are determined by the relation to other execution units with the same cycle time. The maximum value is determined by the number of possible places with the same cycle time. The system selects a free place with the lowest number if no value is specified at entry.

Terminal Description

No	Name	Type	Description
1	ON	IB	Control input which is to be set (to 1) with normal execution.
2	R	IB	**R**eset. Clearing of the execution units of the PC program.
5	RUN	OB	**RUN** indicates that the execution units in the PC program may be executed.

Function

A PC program is the highest level in the PC program structure. The program is intended to perform a complete control function. The program header PCPGM is used to control the enabling, disabling and resetting of the complete PC program, irrespective of subordinate execution units The call parameter C1 specifies how often the program header is to be executed and the call parameter C2 is used to specify the position in the cycle time table on which the execution order of the execution units with the same cycle time is based.

PCPGM must be given a cyclicity from the table for the interpreter with the highest priority. This is required for changes at the inputs to PCPGM to have a direct effect on subordinate execution units. The program header cannot have directly subordinate elements. The control inputs ON and R are used to control when and how the complete PC program with subordinate execution units is to be executed. PCPGM itself is always executed, irrespective of the inputs ON and R.

Normal Execution

The input ON must be set (to 1) for the normal execution of subordinate execution units to be possible. At the start of the execution of subordinate execution units, a check is performed to ensure that the program header has RUN status.

Clearing

If the input R is set, the complete PC program is executed in reset mode. This means that all outputs on the elements in the slave modules are given initial values, which in most cases are the 0-value of the data concerned. Reading to the data base is performed with data value 0 for all data types. Input R overrides input ON.

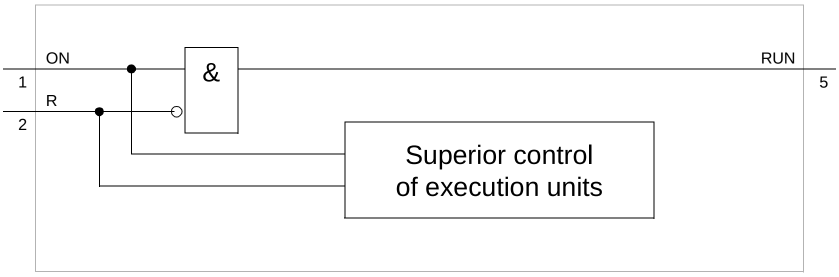
RUN

Output RUN is set only if normal execution is permissible, that is, if the ON input is set and the R input is reset (to 0).

Blocking

A program with subordinate execution units can be blocked in several different ways from a service aid.

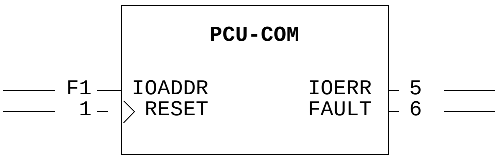
- The command BL<module identity>(**BL**ock) blocks all execution of the PC program except the program header, but inhibits all data exchange with the data base.
- The command BL DB,<module identity>(**BL**ock **D**ata **B**ase) blocks reading to the data base of the data base variables which occur in the PC program.
- The command BLRS<module identity>(**BL**ock and **R**e**S**et) first clears the complete PC program (Reset execution) and then blocks all execution of the program.
- The command BLRS DB,<module identity>(**BL**ock and **R**e**S**et **D**ata **B**ase) first clears the output data to the data base for the complete PC program and then blocks all further reading.
- The command DBL<module identity>(**D**e**BL**ock) is used to cancel blockings.



Function diagram

Pulse Counter, Common Control PCU-COM

Summary



PC element PCU-COM

PCU-COM (**P**ulse **C**ounter **U**nit - **COM**mon Control) is used together with PCU-I, PCU-O and PCU-SS to read and control the advanced pulse counter board DSDP 170. PCU-COM is used for initiation and supervision of DSDP 170 board.

Call PCU-COM

Terminal Description

No	Name	Type	Description
F1	IOADDR	FI	**IOADDR**ess. Strapped I/O address on DSDP 170 in decimal code. The board must be strapped within the interval for Misc Board which is address 192 to 239.
1	RESET	IB	Soft init of DSDP 170 at positive transition. In normal applications, this input is not used. If reset is used, make sure that PCU-O is executed with the same or a shorter cycletime than PCU-COM.
5	IOERR	OB	**I/O ERR**or. Output set when a failed access to DSDP 170 occurs.
6	FAULT	OB	The DSDP 170 board is faulty (F-led=1).

Function

DSDP 170 features four independent speed and position measurements, by counting pulses from incremental pulse generators.

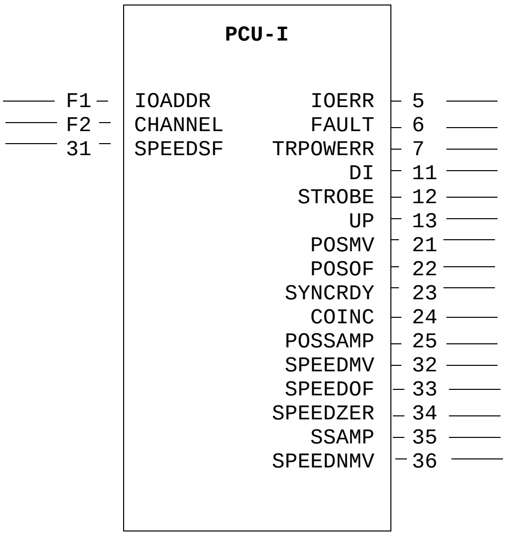
PCU-COM handles the initiation and supervision of DSDP 170 such as transferring basic data to the board and update the System status function in Advant Station 500 OS.

Pulse Counter, Input PCU-I

Summary

PCU-I (**P**ulse **C**ounter **U**nit - **I**n) is used together with PCU-O, PCU-COM and PCU-SS to read and control the advanced pulse counter board DSDP 170. PCU-I is used to read measured values individually for one of the four transducer channels on the board.

Call PCU-I



PC element PCU-I

Terminal Description

No	Name	Type	Description
F1	IOADDR	FI	**IOADDR**ess. Strapped I/O address on DSDP 170 in decimal code. The board must be strapped within the interval for Misc Board which is address 192 to 239.
F2	CHANNEL	FI	**CHANNEL**. Channel (1 - 4) to be used on DSDP 170.
5	IOERR	OB	**I/O ERR**or. Output set when a failed access to DSDP 170 occurs.
6	FAULT	OB	**FAULT**. The DSDP 170 board is faulty (F-led=1).
7	TRPOWERR	OB	**TR**ansducer **POW**er **ERR**or. Supervision of 24V transducer supply.
11	DI	OB	**D**igital Input. State of digital input.
12	STROBE	OB	**STROBE**. Current status of strobe.
13	UP	OB	**UP** direction.
21	POSMV	OIL	POSition Measure Value. Output holding the actual or the frozen (when POSSAMP is true) value of the position counter. Range: ± 8388607
22	POSOF	OB	**POS**ition **O**ver **F**low. True during one execution cycle if POSMV swaps around.
23	SYNCRDY	OB	**SYNC**hronization **R**ead**Y**. Output set when synchronization has occurred. Reset by PCU-O.
24	COINC	OB	**COINC**idence. Output set at coincidence.
25	POSSAMP	OB	**POS**ition **SAMP**led value. The value of POSMV is frozen.

Terminal Description

No	Name	Type	Description
31	SPEEDSF	IR	**SPEED S**cale **F**actor. Scales SPEEDMV from Hz to engineering units. SPEEDSF=0 results in no scaling. Algorithm: If SPEEDSF=0 then SPEEDMV=<frequency in Hz> Else SPEEDMV=<frequency in Hz>/ SPEEDSF
32	SPEEDMV	OR	**SPEED M**easure **V**alue. See also SPEEDSF.
33	SPEEDOF	OB	**SPEED O**ver **F**low. Set if pulse counter for length is full. When SPEEDOF is true the output SPEEDMV is not updated.
34	SPEEDZER	OB	**SPEED ZER**o. Set if pulse counter for time is full. When SPEEDZER is true the output SPEEDMV is 0.
35	SSAMP	OB	**SPEED SAMP**le. SPEEDMV is frozen
36	SPEEDNMV	OB	**SPEED N**ew **M**easurements **V**alue. New speed value has been produced since last read. If SPEEDZER=1, this output is 0, otherwise normally 1.

Function

The values from one channel of DSDP 170 are converted and transferred by PCU-I to the outputs of the element.

DSDP 170 features four independent speed and position measurements, by counting pulses from incremental pulse generators. Board channel 2 and 4 can be defined to synchronize and sample the position counter depending on coincidence at previous channel (1or 3).

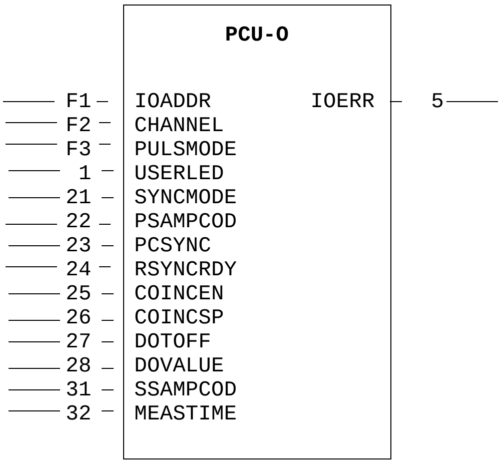
The speed value SPEEDMV is calculated by a division of the number of pulses in the pulse counter with the elapsed time measured by the time counter. The speed is then, if wanted, scaled from Hz to another engineering unit by a division with SPEEDSF. If for example the variable SPEEDMV is wanted in RPM, the scale factor SPEEDSF should be set to: <number of pulses per revolution>/ 60. The board performs continuous speed measurements with the sample time defined by PCU-O connection MEASTIME.

Pulse Counter Output PCU-O

Summary

PCU-O (**P**ulse **C**ounter **U**nit - **O**ut) is used together with PCU-I, PCU-COM and PCU-SS to read and control the advanced pulse counter board DSDP 170. PCU-O is used to give setpoints, orders and parameters individually for one of the four transducer channels on the board.

Call PCU-O



PC element PCU-O

Terminal Description

No	Name	Type	Description
F1	IOADDR	FI	**IOADDR**ess. Strapped I/O address on DSDP 170 in decimal code. The board must be strapped within the interval for Misc Board which is address 192 to 239.
F2	CHANNEL	FI	**CHANNEL.** Channel (1 - 4) to be used on DSDP 170.
F3	PULSMODE	OB	**PULS**e **MODE**. 0: Up/ down pulses from transducer. 1: Quadrature mode, counting positive transitions of CHA. 2: Quadrature mode, counting positive and negative transitions of CHA. 3: Quadrature mode, counting positive and negative transitions of CHA and CHB.
1	USERLED	IB	**USER L**ight **E**mitter **D**iode. Controls the user defined LED at the front of the board.
5	IOERR	OB	**I/O ERR**or. Output set when a failed access to DSDP 170 occurs.
21	SYNCMODE	II	**SYNC MODE.** Choice of synchronization for position counter. 0: Sync inhibit. 1: Sync on PCSYNC 2: Sync on STROBE and UP direction. 3: Sync on STROBE and DOWN direction. 4: Sync on DI+ and UP direction. 5: Sync on DI+ and DOWN direction. 6: Sync on DI– and UP direction. 7: Sync on DI– and DOWN direction. 8: Sync on STROBE and DI+ and UP direction. 9: Sync on STROBE and DI+ and DOWN direction. 10: Sync on STROBE and DI– and UP direction. 11: Sync on STROBE and DI– and DOWN direction. 12: Sync on COINCidence. 13: Sync on COINCidence at previous channel for channel 2 or 4

Terminal Description

No	Name	Type	Description
22	PSAMPCOD	II	**P**osition **SAMP**ling **COD**e 0: Sampling for positive transition at DI. 1: Sampling for negative transition at DI 2: Sampling when coincidence on own channel (for channel 1 and 3), or on previous channel (for channel 2 and 4) occurs. 3: Sampling when order is given from the simultaneous sampling element PCU-SS.
23	PCSYNC	IB	**PC SYNC**hronization. Synchronization of position counter at positive transition if ”SYNCMODE”=1.
24	RSYNCRDY	IB	**R**eset **SYNC R**ea**DY**. Reset sync ready flag at positive transition.
25	COINCEN	IB	**COINC**idence **EN**able. Enable the coincidence detection. If COINC=0 the value of DOVALUE is forced to the digital output.
26	COINCSP	IIL	**COINC**idence **S**et **P**oint. Compared value for coincidence evaluation. limited by element to 3 byte integer (+–8388607).
27	DOTOFF	II	**DO T**ime **O**ff. Duration of the digital output pulses in ms. Limited by element between 1 ms and 16 s. The accuracy of this value is 250µs.
28	DOVALUE	IB	**DO VALUE.** The forced value to the digital output, when COINCEN is false.
31	SSAMPCOD	II	**S**peed **SAMP**ling **COD**e. 0: Sampling for positive transition at DI. 1: Sampling for negative transition at DI. 2: Sampling when coincidence occurred. 3: Sampling when order is given from the simultaneous sampling element PCU-SS.
32	MEASTIME	II	**MEAS**ure **TIME**. Duration of a speed measure period in ms. Limited by element between 1 and 2000 ms. A recommended value for a normal speed measurement is the cycletime of PCU-I divided by two.

Function

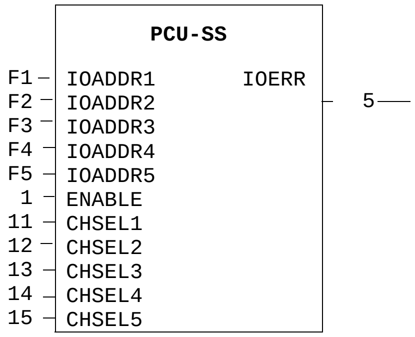
DSDP 170 features four independent speed and position measurements, by counting pulses from incremental pulse generators. PCU-O converts, limits, and transfers the parameters and variables on the input connections to one of the channels of DSDP 170.

Pulse Counter Simultaneous Sampling PCU-SS

Summary

PCU-SS (**P**ulse **C**ounter **U**nit - **S**imultaneous **S**ampling) is used together with PCU-I, PCU-O and PCU-COM to read and control the advanced pulse counter board DSDP 170. PCU-SS is used when simultaneous sampling of several channels for up to five boards is required.

Call PCU-SS



PC element PCU-SS

Terminal Description

No	Name	Type	Description
F1	IOADDR1	FI	**IOADDR**ess board **1**. Strapped I/O address on DSDP 170 in decimal code. The board must be strapped within the interval for Misc Board, address 192 to 239, or 0 if the board is not implemented.
F2	IOADDR2	FI	**IOADDR**ess board **2**.
F3	IOADDR3	FI	**IOADDR**ess board **3**.
F4	IOADDR4	FI	**IOADDR**ess board **4**.
F5	IOADDR5	FI	**IOADDR**ess board **5**.
1	ENABLE	IB	**ENABLE.** Enables the simultaneous sampling of the channels defined in IOADDRx and CHSELx.
5	IOERR	OB	**I/O ERR**or. Output set when a failed access to one of the defined DSDP 170 occurs.
11	CHSEL1	II	**CH**annel **SEL**ect board **1**. Block word specifying the channels which are going to be simultaneous sampled. Requires code 3 on SSAMPCOD and PSAMPCOD of PCU-O. Bit 0: Speed counter channel 1. Bit 1: Speed counter channel 2. Bit 2: Speed counter channel 3. Bit 3: Speed counter channel 4 Bit 4: Pos counter channel 1. Bit 5: Pos counter channel 2. Bit 6: Pos counter channel 3. Bit 7: Pos counter channel 4. Bit 8-15: Not used
12	CHSEL2	II	**CH**annel **SEL**ect board **2**.
13	CHSEL3	II	**CH**annel **SEL**ect board **3**.
14	CHSEL4	II	**CH**annel **SEL**ect board **4**.
15	CHSEL5	II	**CH**annel **SEL**ect board **5**.

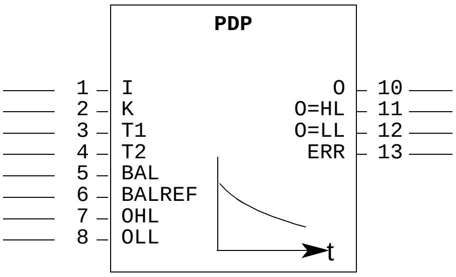
Function

DSDP 170 features four independent speed and position measurements, by counting pulses from incremental pulse generators. PCU-SS handles the coordination to get simultaneous information from up to 20 different channels. PCU-SS is used when this is required, for example, for electrical shaft applications. The number of DSDP 170 boards, treated by PCU-SS is selected by defining the function parameters IOADDRx. If less than five DSDP 170 are used, the remaining addresses are set to zero.

PDP-Function PDP

Summary

PDP (**P**roportional **D**erivating **P**roportional function) is used to give proportional effect and limited derivation effect. The output signal can be limited with limit values specified at special inputs. The balancing function permits the output signal to follow an external reference and permits a bumpless return to the normal function. All transfers from balancing or from limited output signal are bumpless.



PC element PDP

Call PDP

Terminal Description

No	Name	Type	Description
1	I	IR	**I**nput. Input for actual value.
2	K	IR	Input for setting gain.
3	T1	ITR	**T**ime **1**. Input for time constant for lead.
4	T2	ITR	**T**ime **2**. Input for time constant for lag.
5	BAL	IB	**BAL**ance. Input for activation of following.
6	BALREF	IR	**BAL**ance **REF**erence. Input for reference value when following.
7	OHL	IR	**O**utput **H**igh **L**imit. Input for upper limit value.
8	OLL	IR	**O**utput **L**ow **L**imit. Input for lower limit value.
10	O	OR	**O**utput. Output signal.
11	O=HL	OB	**O**utput=**H**igh **L**imit. Output set to 1 if the output O is limited to the upper limit value.
12	O=LL	OB	**O**utput=**L**ow **L**imit. Output set to 1 if the output O is limited to the lower limit value.
13	ERR	OB	**ERR**or. Output set to 1 if OHL is less than OLL.

Function

The step response for a PDP function is:

$O(t)=I(t)\ (K+k1e^{-t/T2})$

where

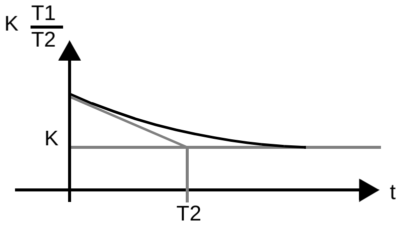
k1=K(T1/T2-1)

The transfer function for a PDP function is:

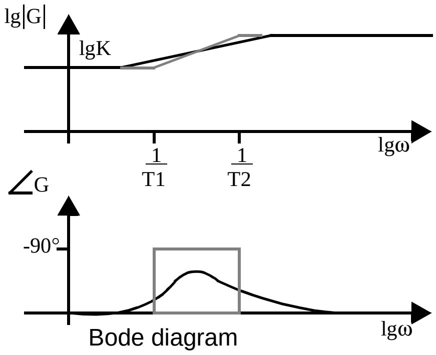
G(s)=K(1+sT1)/(1+sT2)

where

T2<T1



Step response

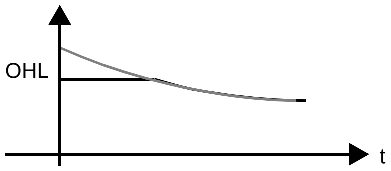


Bode diagram

This has been implemented in the PDP element as a recursive algorithm. The design of this algorithm is such that normal functioning is maintained even during limiting. This ensures a controlled return to a dynamic state.

Gain, Time Constants and Sampling Time

Certain constants are precalculated to make the execution time of the element as short as possible. The results are stored internally in the element. These constants are recalculated if T1, T2 or K are changed by more than 1/128 of their previous value or if the sampling time TS is changed. When recalculating a test is performed to check whether T1 and T2≥2 x TS. If not, T1 or T2 respectively is set equal to 2 x TS.



Function in limiting state

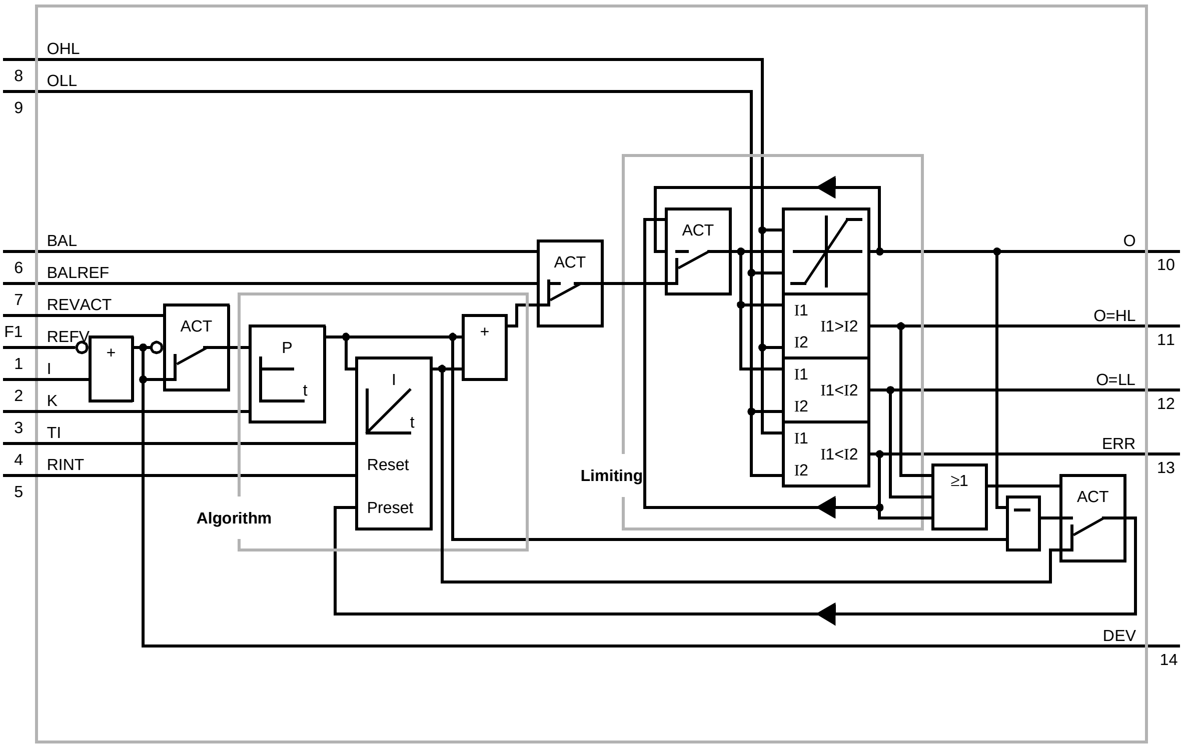
Following

If BAL is set to 1, the regulator immediately goes into following and the output O is set to the value of the input BALREF. If the value at BALREF exceeds the output signal limits, the output is set to the limits value concerned. Return to dynamic state is bumpless.

Limitation Function

The limitation function limits the output signal to the limit values at the inputs OHL for the upper limit and OLL for the lower limit. If the actual value exceeds the upper limit, the output O=HL is set to 1 and if it falls below the lower limit, the output O=LL is set to 1.

The element checks that the upper limit value OHL is greater than the lower limit value OLL. If not, the output ERR is set to 1. While the error status persists, the outputs O=HL, O=LL and O retain the value they had in the sample before, in which the error occurred. After an error the return to a dynamic state is bumpless, in the same way as in the case of following.

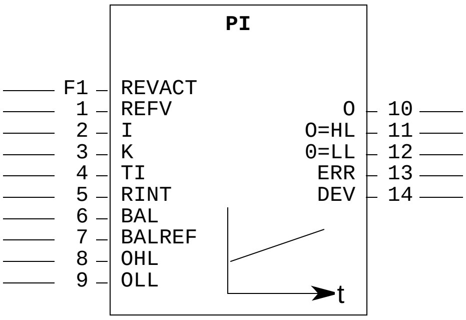


Function diagram

PI-Regulator PI

Summary

PI (**P**roportional **I**ntegrating regulator) is used as a standard PI- regulator for serial compensation in feed back systems. The control deviation is calculated internally in the element. The output signal can be limited to limits specified at special inputs. The balancing function permits the output signal to follow an external reference and permits a bumpless return to normal function. All transfers from balancing or limited output signal are bumpless.



PC element PI

Call PI

Terminal Description

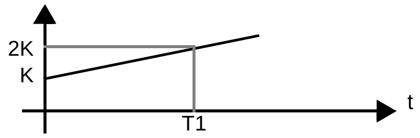
No	Name	Type	Description
F1	REVACT	IB	**REV**erse **ACT**ion. Parameter input for inverted control action. REVACT=1 gives inverted control action.
1	REFV	IR	**REF**erence **V**alue. Input for desired value.
2	I	IR	**I**nput. Input for actual value.
3	K	IR	Input for setting gain.
4	TI	ITR	**T**ime **I**ntegration. Input for time constant for integration.
5	RINT	IB	**R**eset **INT**egrator. Input for clearing of integrator.
6	BAL	IB	**BAL**ance. Input for activation of following.
7	BALREF	IR	**BAL**ance **REF**erence. Input for reference value when following.
8	OHL	IR	**O**utput **H**igh **L**imit. Input for upper limit.
9	OLL	IR	**O**utput **L**ow **L**imit. Input for lower limit.
10	O	OR	**O**utput. Output signal.
11	O=HL	OB	**O**utput=**H**igh **L**imit. Output set to 1 if the output O is limited to the upper limit value.
12	O=LL	OB	**O**utput=**L**ow **L**imit. Output set to 1 if the output is limited to the lower limit value.
13	ERR	OB	**ERR**or. Output set to 1 if OHL is less than OLL.
14	DEV	OR	**DEV**iation. Output for control deviations, that is, DEV=1–REFV.

Function, Control Deviation

DEV, the control deviation, is calculated as follows even when the regulator is following: DEV=I–REFV

Inverted Control Effect

REVACT is a function parameter which controls the direction of the output signal change in relation to the actual value.
If REVACT is set to 0, direct action is obtained, that is, increasing actual value gives increasing output signal. When REVACT is set to 1, inverted action is obtained, that is, increasing actual value gives decreasing output signal.



Step response

Transfer Function

The PI function can be written in the time plane as

$O(t)=K(DEV(t)+\int^{\tau} DEV(\tau)/TI\ d\ \tau)$

and its main property when controlling is that it permits the integral section to retain its value when the error DEV(t)=0, that is, the output signal is then constant.

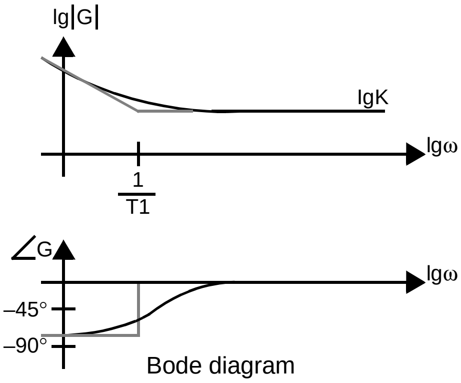
The transfer function for a PI function is:

$G(s)=K(1+1/sTI)$

This has been implemented in the PI element as a recursive algorithm.

Gain, Integration Time Constant and Sampling Time

The constant K x TS/TI is precalculated to make the execution time of the element as short as possible. The result is stored internally in the element. This constant is recalculated if TI or K is changed more than 1/128 of the preceding value or if the sampling time TS is changed. When the value is recalculated a test is performed to ensure that TS/TI<1. If not, TS/TI is set to 1.



Bode diagram

Clearing of Integrator

The integration section of the algorithm is cleared when RINT goes to 1. If the proportional section of the algorithm exceeds the limits of the output signal, the limit status remains and the internal status is updated in accordance with the section "Bumpless transfer from following or limitation". When RINT is set permanently to 1, the element functions as a P-regulator.

Following

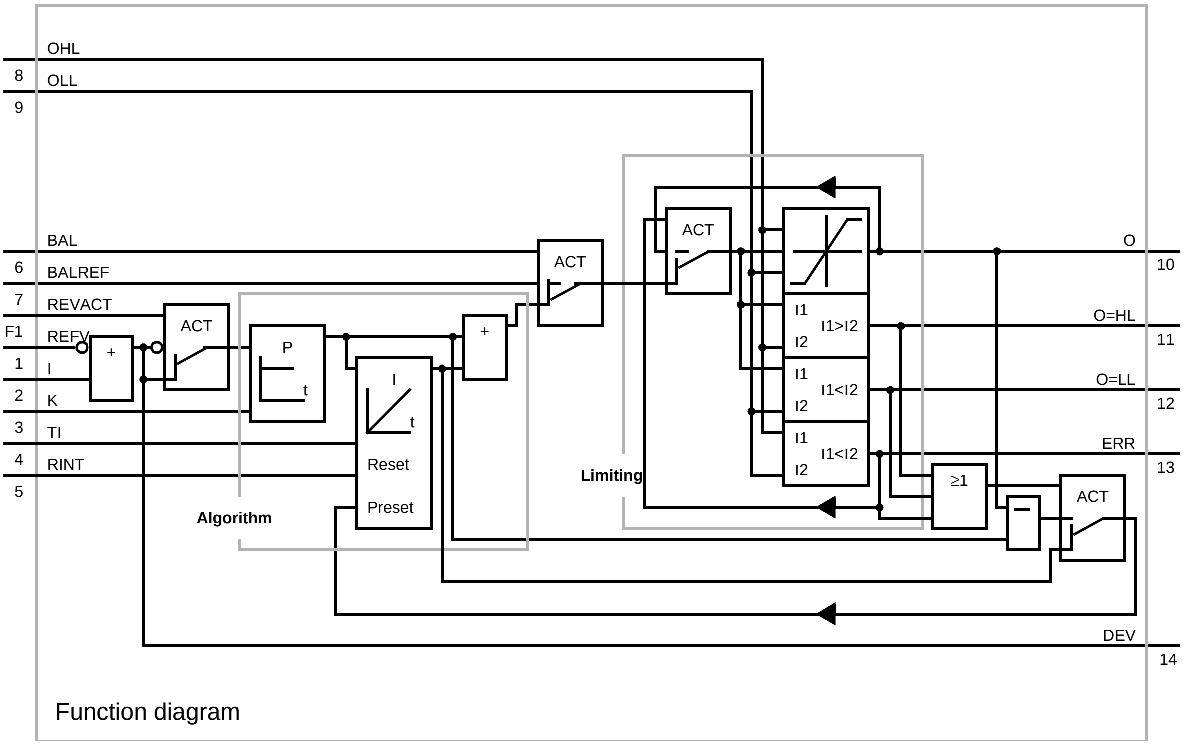
If BAL is set to 1, the regulator immediately goes into following and the output O follows the value of the input BALREF. If the value at BALREF exceeds the output signal limits, the output is set to the limit value concerned. On return to a normal function the value of the output O during the last sample in the following remains during one sample time. See below under "Bumpless transfer from following or limiting".

Limitation Function

The limitation function limits the output signal to the limit values at the inputs OHL for upper limit value and OLL for the lower limit value. If the actual value exceeds the upper limit value, the output O=HL is set to 1 and if it falls below the lower limit value, the output O=LL is set to 1. The element checks that the upper limit value OHL is greater than the lower limit value OLL. If not, the output ERR is set to 1. While the error status persists, the outputs O=HL, O=LL and O retain the values they had in the sample before, in which the error occurred.

Bumpless Transfer from Following or Limitation

Transfer from following status (BAL=1) or from a limited output signal is bumpless. This is performed by recalculation of internal states, that is, the integration part according to INT(t)=O(t) - K x DEV(t).

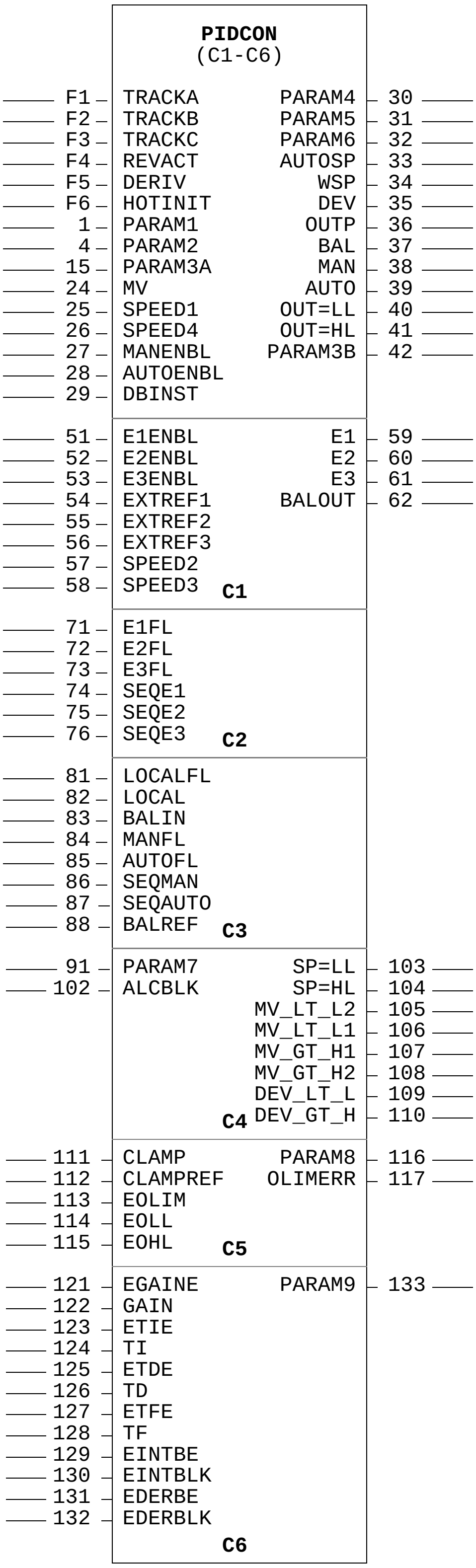


Function diagram

Process Controller PIDCON

Summary

The PC element PIDCON forms part of the functional unit PIDCON (**PID CON**troller). The unit is a flexible and powerful process controller with standardized operator functions. It is used in traditional instrumentation when a VDU based operator station is specified. Further information can be found in the manual *Functional Units part 4, PIDCON, RATIOSTN, MANSTN*



Call PIDCON (C1, C2, C3, C4, C5, C6)

The element consists of one basic part that gives a regulator function in its basic form. This part may be extended with optional functions which are selected using call parameters C1–C6, see the table below.

Parameter	Significance	Permissible values
C1	Ordering of operation mode E1, E2 or E3.	0 - 1
C2	Ordering of operation mode via PC connection.	0 - 1
C3	Ordering of operation mode via PC connection.	0 - 1
C4	Set point value limitation and alarm limits	0 - 1
C5	Output signal control.	0 - 1
C6	Setting of control parameters via PC connection.	0 - 1

Terminal Description

No	Name	Type	Description
F1	TRACKA	II	**TRACK**ing **A**. Function parameter for tracking A.
F2	TRACKB	II	**TRACK**ing **B**. Function parameter for tracking B.
F3	TRACKC	II	**TRACK**ing **C**. Function parameter for tracking C.
F4	REVACT	II	**REV**erse **ACT**ion. Function parameter for control action. Reverse action means that the output signal decreases with increasing measured value. Direct action means that the output signal increases with increasing measured value.
F5	DERIV	II	**DERIV**ation. Function parameter for selection of whether D action is performed on the measured value (MV) or on the deviation (DEV). F5=0 yields D action on the measured value (MV). F5=1 yields D action on the deviation (DEV).
F6	HOTINIT	II	**HOT INIT**. Function parameter for warm start.
1	PARAM1	II	**PARAM**eter **1**. Connection to the data base element (PARAM1).
4	PARAM2	IA48	**PARAM**eter **2**. Connection to the data base element (PARAM2).
15	PARAM3A	II	**PARAM**eter **3 A**. Connection to the data base element (PARAM3).
24	MV	IR	**M**easured **V**alue. Input for measured value.
25	SPEED1	IR	Input for the largest change rate for the set point in AUTO mode (engineering unit/second).
26	SPEED4	IR	Input for the largest change rate for the output in MAN mode. (percent/second).
27	MANENBL	IB	**MAN**ual **EN**a**BL**e. Input for enabling ordering of operation mode MAN by the operator. 0=MAN may not be ordered by the operator. 1=MAN may be ordered by the operator.
28	AUTOENBL	IB	**AUTO EN**a**BL**e. Input for enabling ordering of operation mode AUTO by the operator. 0=AUTO may not be ordered by the operator. 1=AUTO may be ordered by the operator.
29	DBINST	II	**D**ata **B**ase **INST**ance. An operational parameter or a constant, defining the instance number of the data base (to be found in its header). Alternatively, a connection to the data base element.
30	PARAM4	OI	**PARAM**eter **4**. Connection to the data base element (PARAM4).
31	PARAM5	OIL	**PARAM**eter **5**. Connection to the data base element (PARAM5)
32	PARAM6	OA32	**PARAM**eter **6**. Connection to the data base element (PARAM6).

Terminal Description

No	Name	Type	Description
33	AUTOSP	OR	**AUTO S**et **P**oint. Output for set point in operation mode AUTO.
34	WSP	OR	**W**orking **S**et **P**oint. Output for working set point.
35	DEV	OR	**DEV**iation. Output for control deviation.
36	OUTP	OR	**OUT**put. Output signal of the controller, the manipulated variable.
37	BAL	OB	**BAL**ance. Output for operation mode indication. This output is set to 1 with operation mode BAL.
38	MAN	OB	**MAN**ual. Output for operation mode indication. This output is set to 1 with operation mode MAN.
39	AUTO	OB	**AUTO**. Output for operation mode indication. This output is set to 1 with operation mode AUTO.
40	OUT=LL	OB	**OUT**put**=L**ow **L**imit. Output for indication of output signal limitation. It is set to 1 when the output signal equals the lower limit value.
41	OUT=HL	OB	**OUT**put**=H**igh **L**imit. Output for indication of output signal limitation. It is set to 1 when the output signal equals the upper limit value.
42	PARAM3B	OI	**PARAM**eter **3B**. Connection to the data base element (PARAM3). Acknowledgement of switching of operation mode by the operator.
51(1)	E1ENBL	IB	**E1 EN**a**BL**e. Input for enabling ordering of operation mode E1 by the operator. 0=E1 may not be ordered by the operator. 1=E1 may be ordered by the operator.
52(1)	E2ENBL	IB	**E2 EN**a**BL**e. Input for enabling ordering of operation mode E2 by the operator. 0=E2 may not be ordered by the operator. 1=E2 may be ordered by the operator.
53(1)	E3ENBL	IB	**E3 EN**a**BL**e. Input for enabling ordering of operation mode E3 by the operator. 0=E3 may not be ordered by the operator. 1=E3 may be ordered by the operator.
54(1)	EXTREF1	IR	**EXT**ernal **REF**erence **1**. Input for external reference (setpoint) in operation mode E1.
55(1)	EXTREF2	IR	**EXT**ernal **REF**erence **2**. Input for external reference (setpoint) in operation mode E2.
56(1)	EXTREF3	IR	**EXT**ernal **REF**erence **3**. Input for external reference for the output OUT, in operation mode E3.
57(1)	SPEED2	IR	Input for the maximum change rate for the setpoint in mode E1 (engineering unit/second).
58(1)	SPEED3	IR	Input for the maximum change rate for the setpoint in mode E2 (engineering unit/second).
59(1)	E1	OB	Output for operation mode indication. This output is set to 1 with operation mode E1.
60(1)	E2	OB	Output for operation mode indication. This output is set to 1 with operation mode E2.
61(1)	E3	OB	Output for operation mode indication. This output is set to 1 with operation mode E3.
62(1)	BALOUT	OB	**BAL**ance **OUT**. Output for operation mode indication. This output is set to 1 with operation modes BAL, MAN and AUTO.
71(2)	E1FL	IB	**E1 FL**ash. Input for ordering of operation mode E1. It also gives an event report.
72(2)	E2FL	IB	**E2 FL**ash. Input for ordering of operation mode E2. It also gives an event report.
73(2)	E3FL	IB	**E3 FL**ash. Input for ordering of operation mode E3. It also gives an event report.
74(2)	SEQE1	IB	**SEQ**uential **E1**. Pulse input for ordering of operation mode E1.
75(2)	SEQE2	IB	**SEQ**uential **E2**. Pulse input for ordering of operation mode E2.

Terminal Description

No	Name	Type	Description
76(2)	SEQE3	IB	**SEQ**uential **E3**. Pulse input for ordering of operation mode E3.
81(3)	LOCALFL	IB	**LOCAL FL**ash. Input for ordering of operation mode BAL. It gives an event report and VDU indication BAL LOCAL.
82(3)	LOCAL	IB	**LOCAL**. Input for ordering of operation mode BAL. It gives VDU indication BAL LOCAL.
83(3)	BALIN	IB	**BAL**ance **IN**put. Input for ordering of operation mode BAL.
84(3)	MANFL	IB	**MAN**ual **FL**ash. Input for ordering of operation mode MAN. It gives an event report.
85(3)	AUTOFL	IB	**AUTO FL**ash. Input for ordering of operation mode AUTO. It gives an event report.
86(3)	SEQMAN	IB	**SEQ**uential **MAN**ual. Pulse input for ordering of operation mode MAN.
87(3)	SEQAUTO	IB	**SEQ**uential **AUTO**. Pulse input for ordering of operation mode AUTO.
88(3)	BALREF	IR	**BAL**ance **REF**erence. Input for reference value with operation mode BAL.
91(4)	PARAM7	IA48	**PARAM**eter **7**. Connection to the data base element (PARAM7).
102(4)	ALCBLK	IB	**AL**arm **C**ontrol **BL**oc**K**. Input which, when set to 1, blocks limit value alarms for MV and DEV.
103(4)	SP=LL	OB	**S**et **P**oint=**L**ow **L**imit. Output set to 1 when the set point equals the lower limit value.
104(4)	SP=HL	OB	**S**et **P**oint=**H**igh **L**imit. Output set to 1 when the set point equals the upper limit value.
105(4)	MV_LT_L2	OB	**M**easured **V**alue_**L**ess **T**han_**L**ow **2**. Output set to 1 when the measured value (MV) falls below alarm limit L2.
106(4)	MV_LT_L1	OB	**M**easured **V**alue_**L**ess **T**han_**L**ow **1**. Output set to 1 when the measured value (MV) falls below alarm limit L1.
107(4)	MV_GT_H1	OB	**M**easured **V**alue_**G**reater **T**han_**H**igh **1**. Output set to 1 when the measured value exceeds alarm limit H1.
108(4)	MV_GT_H2	OB	**M**easured **V**alue_**G**reater **T**han_**H**igh **2**. Output set to 1 when the measured value exceeds alarm limit H2.
109(4)	DEV_LT_L	OB	**DEV**iation_**LT**_**L**ow. Output set to 1 when the deviation (DEV) falls below the low alarm limit (L).
110(4)	DEV_GT_H	OB	**DEV**iation_**LT**_**H**igh. Output set to 1 when the deviation (DEV) exceeds the high alarm limit (H).
111(5)	CLAMP	IB	Input for ordering of operation mode MAN with clamping.
112(5)	CLAMPREF	IR	**CLAMP REF**erence. Clamp value with operation mode MAN with clamping.
113(5)	EOLIM	IB	**E**xternal **O**utput **LIM**it. Change signal for output signal limits. 0=from the operator’s keyboard. 1=from the inputs EOLL and EOHL.
114(5)	EOLL	IR	**E**xternal **O**utput **L**ow **L**imit. Input for lower limit for the output signal.
115(5)	EOHL	IR	**E**xternal **O**utput **H**igh **L**imit. Input for upper limit for the output signal.
116(5)	PARAM8	OA8	**PARAM**eter **8**. Connection to the data base element (PARAM8).
117(5)	OLIMERR	OB	**O**utput **LIM**it **ERR**or. Output which, when set to 1, indicates an error in the alarm limits, that is, EOLL>EOHL.

Terminal Description

No	Name	Type	Description
121(6)	EGAINE	IB	**E**xternal **GAIN E**nable. Input for selection of GAIN. 0=from the operator's keyboard. 1=from the external signal input GAIN.
122(6)	GAIN	IR	Input for GAIN.
123(6)	ETIE	IB	**E**xternal **TI E**nable. Input for selection of integration time constant. 0=from the operator's keyboard 1=from the external signal input TI
124(6)	TI	ITR	**T**ime **I**ntegration. Input for integration time constant.
125(6)	ETDE	IB	**E**xternal **TD E**nable. Input for selection of derivation time constant. 0=from the operator's keyboard 1=from the external signal input TD
126(6)	TD	ITR	**T**ime **D**erivation. Input for derivation time constant.
127(6)	ETFE	IB	**E**xternal **TF E**nable. Input for selection of filter time constant. 0=from the operator's keyboard 1=from the external signal input TF
128(6)	TF	ITR	**T**ime **F**ilter. Input for the filter time constant for the derivation function.
129(6)	EINTBE	IB	**E**xternal **INT**egration **B**locking **E**nable. Input for selection of blocking signal for integration. 0=Blocking is done from the operator's keyboard. 1=Blocking is done using the signal EINTBLK.
130(6)	EINTBLK	IB	**E**xternal **INT**egration **B**loc**K**ing. Input for blocking and clearing of the integration function.
131(6)	EDERBE	IB	**E**xternal **DER**ivation **B**locking **E**nable. Input for selection of blocking signal for derivation. 0=Blocking is done from the operator's keyboard 1=Blocking is done using the signal EDERBLK.
132(6)	EDERBLK	IB	**E**xternal **DER**ivation **BL**oc**K**ing. Input for blocking and clearing of the derivation function.
133(6)	PARAM9	OA16	**PARAM**eter **9**. Connection to the data base element (PARAM9).

(1) Available only if C1=1.
(2) Available only if C2=1.
(3) Available only if C3=1.
(4) Available only if C4=1.
(5) Available only if C5=1.
(6) Available only if C6=1.

Selection of parameters for the tracking function of AUTOSP

TRACKA	TRACKB	Description
0	0	No tracking
1	1	AUTOSP is not affected. Changeover to AUTO takes place bumplessly by the working setpoint, WSP, taking on the measured value, MV. Thereafter WSP is ramped to equal AUTOSP with a speed that is determined by SPEED1. ”Tracking” takes place in the operating modes BAL, MAN, E1, E2 and E3.
2	1	AUTOSP is not affected. Changeover to AUTO takes place bumplessly by the working setpoint, WSP, taking on the value of EXTREF1 or EXTREF2, depending on whether operating mode E1 or E2 applies. Thereafter WSP is ramped to equal AUTSP with a speed that is determined by SPEED1. ”Tracking ”takes place in the operating modes E1 and E2.

Selection of parameters for the tracking function of AUTOSP

TRACKA	TRACKB	Description
1	2	AUTOSP tracks the measured value, MV. Changeover to AUTO takes place both without bumps and changes by the working setpoint, WSP, taking on the measured value, MV. Tracking takes place in the operating modes BAL, MAN, E1, E2 and E3.
2	2	AUTOSP tracks the external setpoint EXTREF1 or EXTREF2, depending on whether operating mode E1 or E2 applies. Changeover to AUTO takes place both without bumps and changes by the working setpoint, WSP, taking on the value of the external reference EXTREF1 or EXTREF2 as applicable. Tracking takes place in the operating modes E1 and E2.

Selection of parameters for the tracking function of EXTREF

TRACKC	Description
0	No tracking
1	The external setpoints EXTREF1 and EXTREF2 are not affected. Changeover to operating mode E1 or E2 respectively takes place bumplessly by the working setpoint, WSP, taking on the measured value, MV. Thereafter WSP is ramped to equal EXTREF1 or EXTREF2 as applicable with a speed that is determined by SPEED2 or SPEED3 respectively. ”Tracking” of EXTREF1 takes place in the operating modes BAL, MAN, E2 and E3, while ”tracking” of EXTREF2 takes place in the modes BAL, MAN, E1 and E3.
2	The external setpoints EXTREF1 and EXTREF2 are not affected. Changeover to operating mode E1 or E2 respectively takes place bumplessly by the working setpoint, WSP, retaining its value from the preceding mode. Thereafter WSP is ramped to equal EXTREF1 or EXTREF2 as applicable with a speed that is determined by SPEED2 or SPEED3 respectively. ”Tracking” of EXTREF1 takes place in the operating modes BAL, MAN, E2 and E3, while ”tracking” of EXTREF2 takes place in the modes BAL, MAN, E1 and E3.

REVACT

REVACT	Function PI-part	Function D- part
0	direct	direct
1	reverse	reverse
2	direct	reverse
3	reverse	direct

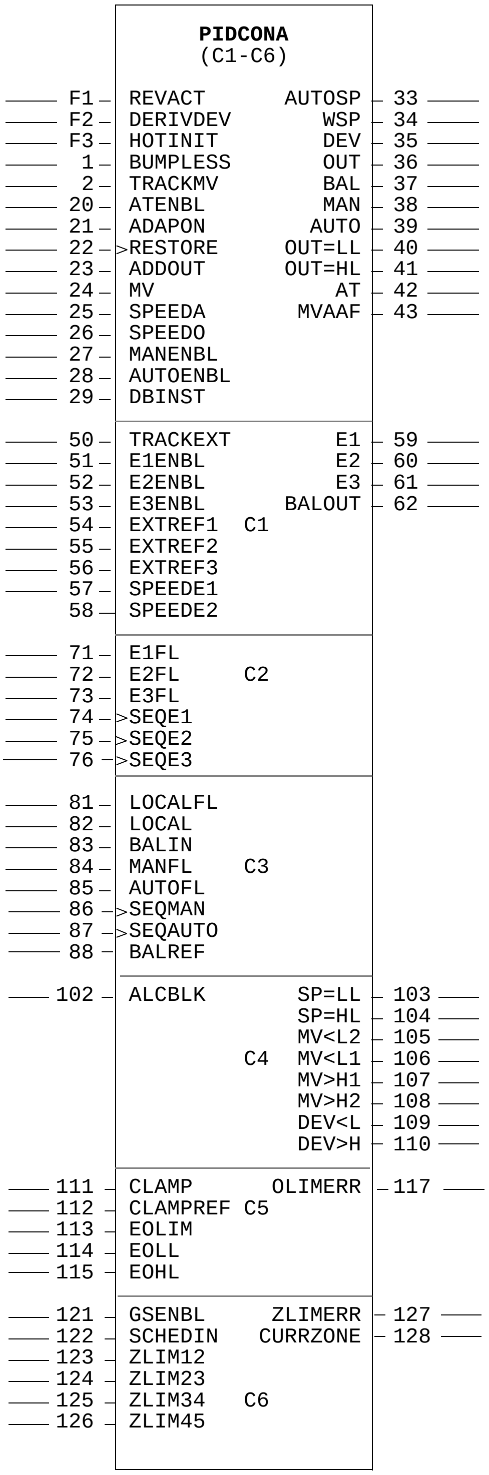
HOTINIT

Function on warm start		
HOTINIT	**Mode**	**Output**
0	unchanged	unchanged
1	MAN	unchanged
2	MAN	0 or OLL
Unchanged means: no change in relation to the state in effect before the event that caused the warm start.		

Process Controller with Autotuning PIDCONA

Summary

The PC element PIDCONA forms part of the functional unit PIDCONA (**PID CON**troller with built in **A**utotuning and adaptive control). The unit is a flexible and powerful process controller with standardized operator functions. It is used in traditional instrumentation when a VDU based operator station is specified. Further information can be found in the manual *Functional Units part 7, PIDCONA*



PC Element PIDCONA

Call PIDCONA (C1, C2, C3, C4, C5, C6)

The element consists of one basic part that gives a regulator function in its basic form. This part may be extended with optional functions which are selected using call parameters C1–C6, see table below.

Parameter	Significance	Permissible values
C1	Ordering of operation mode E1, E2 or E3.	0 - 1
C2	Ordering of operation mode E1, E2 or E3 via PC connection	0 - 1 (1 requires C1=1)
C3	Ordering of operation mode BAL, MAN or AUTO via PC connection.	0 - 1
C4	Alarm limits	0 - 1
C5	Output signal control	0 - 1
C6	Gain scheduling	0 - 1

Terminal Description

No	Name	Type	Description
F1	REVACT	II	**REV**erse **ACT**ion. Function parameter for control action. Reverse action (REVACT =1) means that the output signal (OUT) decreases with increasing measured value (MV) (the normal case). Direct action (REVACT=0) means that the output signal increases with increasing measured value (used when a change of the manipulated variable (OUT) leads to a final change of the controlled variable (MV) in the opposite direction).
F2	DERIVDEV	II	**DERIV**ation of the **DEV**iation. Function parameter for selection of whether D action is performed on the measured value (MV) or on the deviation (DEV). DERIVDEV=0 is normally recommended. It yields D action on the measured value (MV). DERIVDEV=1 yields D action on the deviation (DEV).
F3	HOTINIT	II	**HOT INIT**. Function parameter determining the state after a warm start. HOTINIT=0 is chosen, unless special actions are required. HOTINIT=0 gives unchanged mode and unchanged output. HOTINIT=1 gives mode ”MAN” and unchanged output. HOTINIT=2 gives mode ”MAN” and OUT=0. If 0 is outside the limits, OUT is set to the closest limit. By unchanged is meant no change in relation to the state in effect before the event that caused the warm start (the conditions are described in the manual *Functional Units, Part1*).
1	BUMPLESS	IB	**BUMPLESS** transfer. Input for selection of bumpless transfer from mode BAL, MAN or E3 to mode AUTO, E1, or E2. With BUMPLESS=1 there will be no ”P step” on transition, and the working setpoint WSP is ramped from the value giving no output change to the value for the new mode, using that mode's setpoint rate limit. With BUMPLESS=0 the new setpoint will be used directly, without ramping to it, and there will be a ”P step” if there is an initial deviation.
2	TRACKMV	IB	**TRACK** the **M**easured **V**alue. Input for selection of tracking or no tracking for the operator setpoint AUTOSP in the modes BAL, MAN, and E3. With TRACKMV=0, there is no tracking, and AUTOSP is available for operator changes in these modes. With TRACKMV=1, AUTOSP tracks the measured value in these modes, thereby making the transition to mode AUTO free from output change.
20	ATENBL	IB	**A**uto**T**uning **EN**a**BL**e. Input for allowing autotuning. If ATENBL=1, autotuning is allowed. If ATENBL=0, autotuning is interrupted and can not be started.
21	ADAPON	IB	**ADAP**tive control **ON**. Input for requesting the control to be adaptive in modes AUTO, E1, and E2.

Terminal Description

No	Name	Type	Description
22	RESTORE	IB	**RESTORE** saved controller settings. Dynamic input. A positive transition causes the saved tuning (of the current zone, if C6=1) to replace the present one.
23	ADDOUT	IR	**ADD**ed **OUT**put value. Input that is added to the control algorithm output. Used for bias, when there is no integral action, or for feed forward.
24	MV	IR	**M**easured **V**alue. Input for the measured value (the feedback signal representing the controlled variable).
25	SPEEDA	IR	Input for the largest change rate for the setpoint in AUTO mode (engineering unit/second).
26	SPEEDO	IR	Input for the largest change rate for the output (OUT) (engineering unit/second). Applies to all modes except BAL.
27	MANENBL	IB	**MAN**ual **EN**a**BL**e. Input for enabling ordering of operation mode MAN by the operator. 0=MAN may not be ordered by the operator. 1=MAN may be ordered by the operator.
28	AUTOENBL	IB	**AUTO EN**a**BL**e. Input for enabling ordering of operation mode AUTO by the operator. 0=AUTO may not be ordered by the operator. 1=AUTO may be ordered by the operator.
29	DBINST	II	**D**ata **B**ase **INST**ance. An operational parameter or a constant, defining the instance number of the data base (to be found in its header). Alternatively, a connection to the data base element.
33	AUTOSP	OR	**AUTO S**et **P**oint. Output for setpoint in operation mode AUTO.
34	WSP	OR	**W**orking **S**et**P**oint. Output for working setpoint.
35	DEV	OR	**DEV**iation. Output for control deviation.
36	OUT	OR	**OUT**put. Output signal of the controller, the manipulated variable.
37	BAL	OB	**BAL**ance. Output for operation mode indication. This output is set to 1 with operation mode BAL.
38	MAN	OB	**MAN**ual. Output for operation mode indication. This output is set to 1 with operation mode MAN.
39	AUTO	OB	**AUTO**. Output for operation mode indication. This output is set to 1 with operation mode AUTO.
40	OUT=LL	OB	**OUT**put**=L**ow **L**imit. Output for indication of output signal limitation. It is set to 1 when the output signal equals lower limit value.
41	OUT=HL	OB	**OUT**put**=H**igh **L**imit. Output for indication of output signal limitation. It is set to 1 when the output signal equals the upper limit value.
42	AT	OB	**A**uto**T**uning. Output indicating 1 during autotuning, otherwise 0.
43	MVAAF	OR	**M**easured **V**alue, **A**nti-**A**lias **F**iltered. Output for the measured value after anti-alias filtering and sampling, with the rate presently used in the control algorithm.
50(1)	TRACKEXT	II	**TRACK** the **EXT**ernal reference. Input for selection of tracking or no tracking for the operator set point AUTOSP in the modes E1 and E2. With TRACKTEXT=0, there is no tracking, and AUTOSP is available for operator changes in these modes. With TRACKTEXT=1, AUTOSP tracks the working set point WSP. This gives change-free transitions from E1 and E2 to AUTO. With TRACKTEXT=2, AUTOSP tracks EXTREF1 in mode E1, and EXTREF2 in mode E2. Thereby a set point ramp started in E1 or E2 will continue to its final value (with rate SPEEDA) after transition to AUTO.

Terminal Description

No	Name	Type	Description
51(1)	E1ENBL	IB	**E1 EN**a**BL**e. Input for enabling ordering of operation mode E1 by the operator. 0=E1 may not be ordered by the operator. 1=E1 may be ordered by the operator.
52(1)	E2ENBL	IB	**E2 EN**a**BL**e. Input for enabling ordering of operation mode E2 by the operator. 0=E2 may not be ordered by the operator. 1=E2 may be ordered by the operator.
53(1)	E3ENBL	IB	**E3 EN**a**BL**e. Input for enabling ordering of operation mode E3 by the operator. 0=E3 may not be ordered by the operator. 1=E3 may be ordered by the operator.
54(1)	EXTREF1	IR	**EXT**ernal **REF**erence **1**. Input for external reference (setpoint) in operation mode E1.
55(1)	EXTREF2	IR	**EXT**ernal **REF**erence **2**. Input for external reference (setpoint) in operation mode E2.
56(1)	EXTREF3	IR	**EXT**ernal **REF**erence **3**. Input for external reference for the output OUT, in operation mode E3.
57(1)	SPEEDE1	IR	Input for the maximum change rate for the setpoint in mode E1 (engineering unit/second).
58(1)	SPEEDE2	IR	Input for the maximum change rate for the setpoint in mode E2 (engineering unit/second).
59(1)	E1	OB	Output for operation mode indication. This output is set to 1 with operation mode **E1**.
60(1)	E2	OB	Output for operation mode indication. This output is set to 1 with operation mode **E2**.
61(1)	E3	OB	Output for operation mode indication. This output is set to 1 with operation mode **E3**.
62(1)	BALOUT	OB	**BAL**ance **OUT**. Output for operation mode indication. This output is set to 1 with operation modes BAL, MAN and AUTO.
71(2)	E1FL	IB	**E1 FL**ash. Input for ordering of operation mode E1. It also gives an event report.
72(2)	E2FL	IB	**E2 FL**ash. Input for ordering of operation mode E2. It also gives an event report.
73(2)	E3FL	IB	**E3 FL**ash. Input for ordering of operation mode E3. It also gives an event report.
74(2)	SEQE1	IB	**SEQ**uential **E1**. Dynamic input for ordering of operation mode E1.
75(2)	SEQE2	IB	**SEQ**uential **E2**. Dynamic input for ordering of operation mode E2.
76(2)	SEQE3	IB	**SEQ**uential **E3**. Dynamic input for ordering of operation mode E3.
81(3)	LOCALFL	IB	**LOCAL FL**ash. Input for ordering of operation mode BAL. It gives an event report and VDU indication BAL LOCAL.
82(3)	LOCAL	IB	**LOCAL**. Input for ordering of operation mode BAL. It gives VDU indication BAL LOCAL.
83(3)	BALIN	IB	**BAL**ance **IN**put. Input for ordering of operation mode BAL.
84(3)	MANFL	IB	**MAN**ual **FL**ash. Input for ordering of operation mode MAN. It gives an event report.
85(3)	AUTOFL	IB	**AUTO FL**ash. Input for ordering of operation mode AUTO. It gives an event report.
86(3)	SEQMAN	IB	**SEQ**uential **MAN**ual. Dynamic input for ordering of operation mode MAN.
87(3)	SEQAUTO	IB	**SEQ**uential **AUTO**. Dynamic input for ordering of operation mode AUTO.
88(3)	BALREF	IR	**BAL**ance **REF**erence. Input for reference value for the manipulated variable (the output signal, OUT) with operation mode BAL.
102(4)	ALCBLK	IB	**AL**arm **C**ontrol **BL**oc**K**. Input which, when set to 1, blocks limit value alarms for MV and DEV.

Terminal Description

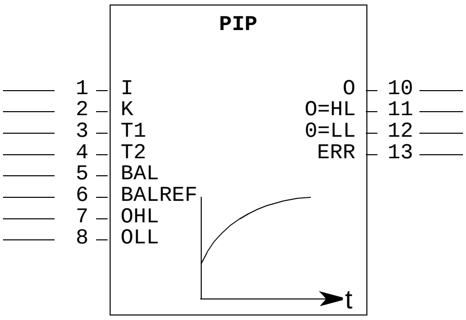
No	Name	Type	Description
103(4)	SP=LL	OB	**S**et**P**oint**=L**ow **L**imit. Output set to 1 when the setpoint equals the lower limit value.
104(4)	SP=HL	OB	**S**et **P**oint**=H**igh **L**imit. Output set to 1 when the set point equals the upper limit value.
105(4)	MV<L2	OB	**M**easured **V**alue<**L**ow **2**. Output set to 1 when the measured value (MV) falls below alarm limit L2.
106(4)	MV<L1	OB	**M**easured **V**alue<**L**ow **1**. Output set to 1 when the measured value (MV) falls below alarm limit L1.
107(4)	MV>H1	OB	**M**easured **V**alue>**H**igh **1**. Output set to 1 when the measured value exceeds alarm limit H1.
108(4)	MV>H2	OB	**M**easured **V**alue>**H**igh **2**. Output set to 1 when the measured value exceeds alarm limit H2.
109(4)	DEV<L	OB	**DEV**iation<**L**ow. Output set to 1 when the deviation (DEV) falls below the low alarm limit (L).
110(4)	DEV>H	OB	**DEV**iation>**H**igh. Output set to 1 when the deviation (DEV) exceeds the high alarm limit (H).
111(5)	CLAMP	IB	**CLAMP** the output (OUT) to its clamp reference value. Input for ordering of operation mode MAN with clamping. The mode and the output can not be changed by the operator in this mode.
112(5)	CLAMPREF	IR	**CLAMP REF**erence. Clamp value with operation mode MAN with clamping.
113(5)	EOLIM	IB	**E**xternal **O**utput **LIM**it. Change signal for output signal limits. 0=from the operator's keyboard. 1=from the inputs EOLL and EOHL.
114(5)	EOLL	IR	**E**xternal **O**utput **L**ow **L**imit. Input for lower limit for the output signal.
115(5)	EOHL	IR	**E**xternal **O**utput **H**igh **L**imit. Input for upper limit for the output signal.
117(5)	OLIMERR	OB	**O**utput **LIM**it **ERR**or. Output which, when set to 1, indicates an error in the alarm limits, that is, EOLL>EOHL.
121(6)	GSENBL	IB	**G**ain **S**cheduling **EN**a**BL**e. Input to select gain scheduling active (GSENBL=1), or turned off (GSENBL=0). When active, saved settings are restored on changes of current zone. When turned off, zone 1 is considered current independent of SCHEDIN. The mere change of GSENBL does not cause any settings to be restored.
122(6)	SCHEDIN	IR	**SCHED**uling **IN**put. Input for the gain scheduling variable. Its value, compared to the zone limits, determines the current zone.
123(6)	ZLIM12	IR	**Z**one **LIM**it between zones **1** and **2**. Input determining this gain scheduling zone limit.
124(6)	ZLIM23	IR	**Z**one **LIM**it between zones **2** and **3**. Input determining this gain scheduling zone limit.
125(6)	ZLIM34	IR	**Z**one **LIM**it between zones **3** and **4**. Input determining this gain scheduling zone limit.
126(6)	ZLIM45	IR	**Z**one **LIM**it between zones **4** and **5**. Input determining this gain scheduling zone limit.
127(6)	ZLIMERR	OB	**Z**one **LIM**it **ERR**or. Output which, when set to 1, indicates an error in the relations between the zone limits. It is required that the limits are ascending, that is, ZLIM12≤ZLIM23≤ZLIM34≤ZLIM45.
128(6)	CURRZONE	OI	CURRent ZONE. Output indicating the current zone of gain scheduling. It equals 1, if gain scheduling is not enabled.

(1) Available only if C1=1.
(2) Available only if C2=1.
(3) Available only if C3=1.
(4) Available only if C4=1.
(5) Available only if C5=1.
(6) Available only if C6=1.

PIP-Function PIP

Summary

PIP (**P**roportional **I**ntegrating **P**roportional function) is used to give a proportional effect and a limited integration effect. The output signal in the steady state is proportional to the input signal. The output signal can be limited with limit values specified at special inputs. The balancing function permits the output signal to follow an external reference and permits a bumpless return to the normal function. All transfers from static states are bumpless.



PC element PIP

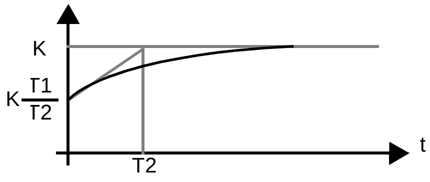
Call PIP

Terminal Description

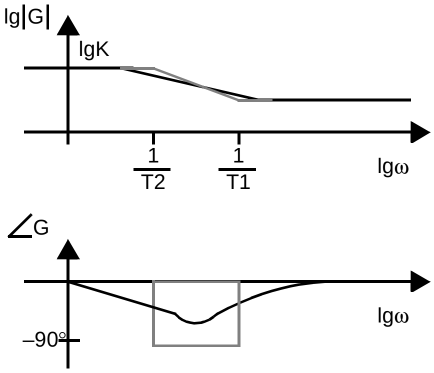
No	Name	Type	Description
1	I	IR	**I**nput. Input for actual value.
2	K	IR	Input for setting gain.
3	T2T1	ITR	**T**ime **1**. Input for time constant for lead.
4	T2	ITR	**T**ime **2**. Input for time constant for lag.
5	BAL	IB	**BAL**ance. Input for activation of following.
6	BALREF	IR	**BAL**ance **REF**erence. Input for reference value when following.
7	OHL	IR	**O**utput **H**igh **L**imit. Input for upper limit value.
8	OLL	IR	**O**utput **L**ow **L**imit. Input for lower limit value.
10	O	OR	**O**utput. Output signal.
11	O=HL	OB	**O**utput=**H**igh **L**imit. Output set to 1 if the output O is limited to the upper limit value.
12	O=LL	OB	**O**utput=**L**ow **L**imit. Output set to 1 if the output O is limited to the lower limit value.
13	ERR	OB	**ERR**or. Output set to 1 if OHL is less than OLL.

Function, Transfer Function

The step response in the time plane for the PIP function is: O(t)=I(t) (K+k1xe^{-}t/T2)
where k1=K (T1/T2-1)
The transfer function for a PIP function is: G(s)=K(1+sT1)/(1+sT2)
where T2>T1
This has been implemented in the PIP element as a recursive function.



Step response



Bode diagram

Gain, Time Constants and Sampling Time

Certain constants are precalculated to make the execution time of the element as short as possible. The result is stored internally in the element. These constants are recalculated if T1, T2 or K are changed by more than 1/128 of their previous value or if the sampling time TS is changed. When recalculating a test is performed to check if T1 and T2≥2 x TS. If not, T1 or T2 respectively is set equal to 2 x TS.

Following

If BAL is set to 1, the regulator immediately goes into following and the output O is set to the value of the input BALREF.
If the value at BALREF exceeds the output signal limits, the output is set to the limit value concerned. Return to dynamic state is bumpless.

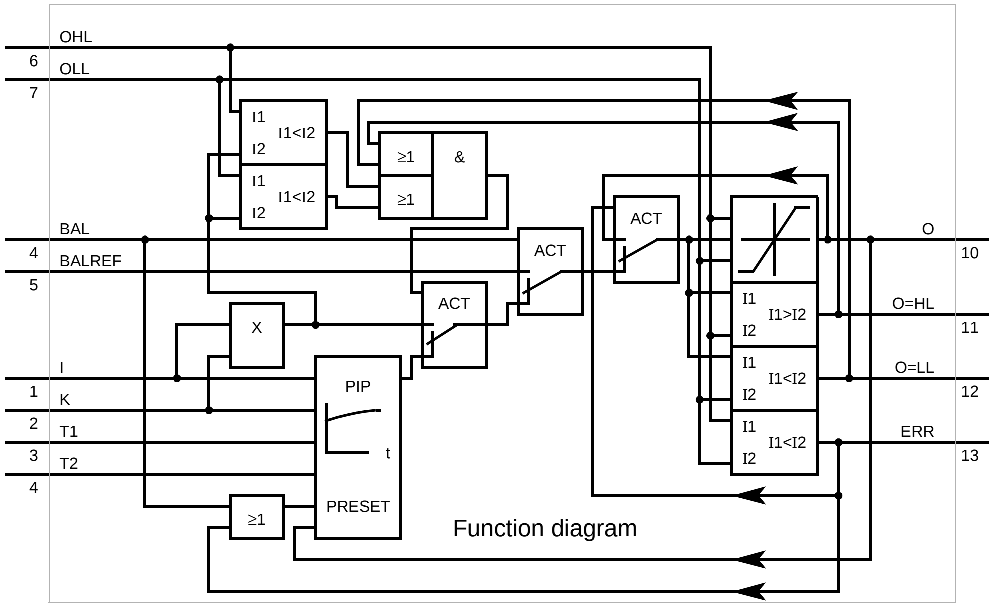
Limitation Function

The limitation functions limits the output signal to the values at the inputs OHL for the upper limit and OLL for the lower limit. If the actual value exceeds the upper limit, the output O=HL is set to 1 and if it falls below the lower limit, the output O=LL is set to 1.

When the limitation status has been detected, a check is made each time the element is executed to determine whether K x I(t) exceeds the output signal limitations. If so, the limitation status remains. If not, the calculation of the output signal is performed by the algorithm in the normal way.

Normal updating of the internal status is sufficient to ensure bumpless return to the dynamic status. The element checks that the upper limit value OHL is greater than the lower limit value OLL. If not, the output ERR is set to 1.

While the error status persists, the outputs O=HL, O=LL and O retain the values they had in the sample before that in which the error occurred. After an error the return to a dynamic state is bumpless, in the same way as in the case of following.

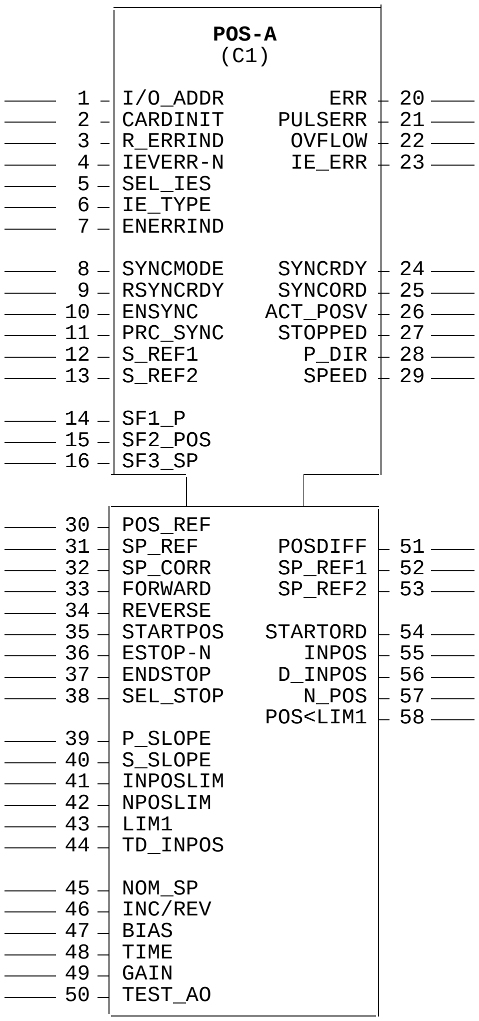


Function diagram

Positioning, Analog Control POS-A

Summary

POS-A (**POS**itioning **- A**nalog control) is used with DSDP 140A for length measurement or position control.



PC element POS-A (1)

Call POS-A (C1)

Parameter	Significance	Permissible values
C1	Selection of control mode C1=0 gives position measurement only. C1=1 gives analog position control.	0 - 1

Terminal Description

No	Name	Type	Description
F1	I/O_ADDR	II	**I/O ADD**ress. Strapped I/O address on DSDP 140A in decimal code. The board must be strapped to an even address within the interval for Misc Board which is address 192 to 239.
2	CARDINIT	IB	**CARD INIT**ialize. Input for starting the board, for example, after TEST_AO.
3	R_ERRIND	IB	**R**eset **ERR**or **IND**ication. Input for clearing of error indication.
4	IEVERR-N	IB	**I**ncremental **E**ncoder **V**oltage **ERR**or **- N**. Input for external signal for supervision of pulse transmitter voltage. 0=voltage too low. 1=normal voltage.
5	SEL_IES	IB	**SEL**ect **I**ncremental **E**ncoder **S**upervision. Input for selection of internal or external supervision of the pulse transmitter. 0=via input IEVERR-N of the PC element. 1=via internal DI on DSDP 140A.
6	IE_TYPE	IB	**I**ncremental **E**ncoder **TYPE**. Input for selection of pulse transmitter type. 0=QGFA 110/120. 1=other types.
7	ENERRIND	IB	**EN**able **ERR**or **IND**ication. Input for selection of error indications via LED on DSDP 140A. 0=normal operation indication. 1=normal operation indication + error indication on LED
8	SYNCMODE	II	**SYNC**hronization **MODE**. Input for selection of synchronization mode.
9	RSYNCRDY	IB	**R**eset **SYNCRDY**. Input for clearing of SYNCRDY. If this input is 0 the entire positioning system is blocked.
10	ENSYNC	IB	**EN**able **SYNC**hronization. Input which, when set to 1, permits synchronization according to condition SYNCMODE.
11	PRC_SYNC	IB	**PR**ogram **C**ontrolled **SYNC**. Pulse input for program controlled synchronization. This synchronization is performed irrespective of ENSYNC, and conditions according to SYNCMODE do not apply.
12	S_REF1	IR	**S**ynchronization **REF**erence **1**. Reference value in engineering unit for synchronization. Selected with signal SYNC 1.
13	S_REF2	IR	**S**ynchronization **REF**erence **2**. Reference value in engineering unit for synchronization. Selected with signal SYNC 2.
14	SF1_P	II	**SF1 P**ulse. Input for scale factor (2 bits). 1: A’s leading edge. 2: A’s and B’s leading edges. 4: A’s and B’s leading and trailing edges.
15	SF2_POS	IR	**SF2 POS**ition. Input for scale factor given as: The selected value of SF1 must be considered in the SF2 value given. The PC element gives a reference scaled in pulses (S-REF x SF2) to the board.
16	SF3_SP	II	**SF3 SP**eed. Input for scale factor for speed.
20	ERR	OB	**ERR**or. Collective error signal.
21	PULSERR	OB	**PULS**e **ERR**or. Output which, when set to 1, indicates an error in the pulse signal.
22	OVFLOW	OB	**OV**er**FLOW**/underflow. Output which, when set to 1, indicates an error in the counter for pos value.
23	IE_ERR	OB	**IE ERR**or. Output which, when set to 1, indicates a fault in the pulse transmitter, cable or signal.

Terminal Description

No	Name	Type	Description
24	SYNCRDY	OB	**SYNC**hronization **R**ea**DY**. Output which, when set to 1, indicates that synchronization has been performed.
25	SYNCORD	OB	**SYNC**hronization **ORD**er. Output which, when set to 1, indicates that synchronization has been ordered. The output remains set to 1 for 5 seconds.
26	ACT_POSV	OR	**ACT**ual **POS**ition **V**alue. Output for position value in engineering unit.
27	STOPPED	OB	Output which, when set to 1, indicates that the object is stationary, that is, the value of the position counter has not changed since the previous reading.
28	P_DIR	OB	**P**ositive **DIR**ection. Output for indication of motion direction. 1: Positive direction. 0: Negative direction.
29	SPEED	OI	Output, the value of which indicates the speed of the object expressed as number of pulses/ 90 ms, scaled with the factor SF3_SP.
30	POS_REF	IR	**POS**ition **REF**erence. Input for positioning reference.
31	SP_REF	II	**SP**eed **REF**erence. Input for external speed reference. The input indicates the current maximum speed. The reference 0-32767 corresponds to 0-100% speed (0-10 V at the analog output).
32	SP_CORR	II	**SP**eed **CORR**ection. Input for external speed correction. The value at this input is added to the output signal after the ramp but before the D/A converter. The reference 0-32767 corresponds to 0-100% speed (0-10 V at the analog output).
33	FORWARD	IB	Input which, when set to 1, gives a positive sign at the output SP_REF1/2.
34	REVERSE	IB	Input which, when set to 1, gives a negative sign at the output SP_REF1/2.
35	STARTPOS	IB	**START POS**itioning. Input for start condition for positioning. The input is internally interlocked by means of the signals FAULT, SYNCRDY, ESTOP-N and DINPOS (if the object is in the correct position).
36	ESTOP-N	IB	**E**mergency **STOP-N**. Input which, when set to 0 interrupts the positioning and blocks the analog output signal.
37	ENDSTOP	IB	**END STOP**. Input for external limit position. The input blocks the analog output signal.
38	SEL_STOP	IB	**SEL**ect **STOP** in position. 0: Continuous position control according to the input POSREF. 1: Selection of stop via the delay DINPOS, that is, the positioning is interrupted when the position corresponds to POSREF.
39	P_SLOPE	IR	**P**ositioning **SLOPE**. Input for the slope of the ramp for n-ref when positioning is given in %/second (per 10 V output signal).
40	S_SLOPE	IR	**S**peed **SLOPE**. Input for the slope of the ramp for n-ref with external n-ref.
41	INPOSLIM	IR	**IN POS**ition **LIM**it. Limit value for indication of correct position. This input indicates the maximum deviation of the real value from the reference for which it is still considered to be in the correct position (INPOS set to 1).
42	NPOSLIM	IR	**N**ear **POS**ition **LIM**it. Input for limit value for POSDIFF (near position).
43	LIM1	IR	**LIM**it **1**. Limit value for position.

Terminal Description

No	Name	Type	Description
44	TD_INPOS	II	**T**ime **D**elay **INPOS**. Input time during which the object must remain in the correct position (POSREF±INPOSLIM) before STARTORD is reset to 0. Given in number of 20 ms intervals, that is, 0-255 x 20 ms.
45	NOM_SP	IR	**NOM**inal **SP**eed. Input for nominal maximum speed or maximum rate of revolution.
46	INC/REV	IR	**INC**rements**/REV**olution. Input giving the number of pulses per revolution from the pulse transmitter.
47	BIAS	IR	Input for bias (least n-ref) in the parable function (rpm).
48	RTIME	IR	**R**ise **TIME**. Input for rise time for revolution rate control in seconds.
49	GAIN	IR	Slope compensation for the parable - normally 1.0. Results from recalculation in the PC element of parameters for the parable generator comprise co-ordinates and slopes.
50	TEST_AO	IB	**TEST A**nalog **O**utput. Input for starting the test of the analog output. The test starts at DIBM PC if the DSDP 140A has not been initialized.
51	POSDIFF	OR	**POS**ition **DIFF**erence. Output for control deviation, that is, POSREFzPOSVALUE.
52	SP_REF1	OI	**SP**eed **REF**erence **1**. Output for speed reference before the ramp.
53	SP_REF2	OI	**SP**eed **REF**erence **2**. Output for speed reference after the ramp.
54	STARTORD	OB	**START ORD**ered. Output for indication that positioning has been ordered via STARTPOS or FORWARD/REVERSE.
55	INPOS	OB	**IN POS**ition. Output that indicates that the set position has been reached, that is, POSDIFF<INPOSLIM.
56	D_INPOS	OB	**D**elayed **INPOS**. Output which, when set to 1, indicates that INPOS has been in effect for the time TDINPOS. It is used to deactivate STARTORD.
57	N_POS	OB	**N**ear **POSi**tion. Indicates that POSDIFF<NPOSLIM.
58	POS<LIM1	OB	**POS**ition**<LIM1**. Indicates that POS<LIM 1.

SYNCMODE

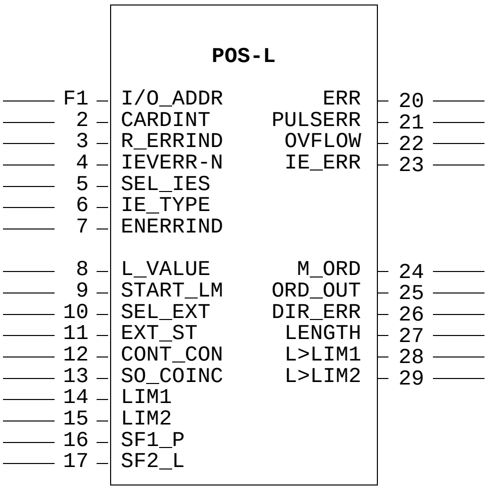
	edge	
Selection of sync conditions	**leading**	**trailing**
SYNC1	0	10
SYNC1 & STROBE	4	-
SYNC1 & STROBE & up counting	5	-
SYNC1 & STROBE & down counting	6	-
SYNC1 & up counting	2	12
SYNC1 & down counting	3	13
STROBE	7	-
SYNC1 or SYNC2	1	-

Length Measurement POS-L

Summary

POS-L (**POS**itioning **- L**ength) is used with DSDP 140A for length measurement with a digital output for coincidence.

Call POS-L



PC element POS-L

Terminal Description

No	Name	Type	Description
F1	I/O_ADDR	II	**I/O ADD**ress. Strapped I/O address on DSDP 140A in decimal code. The board must be strapped to an even address within the interval for Misc Board which is address 192 to 239.
2	CARDINIT	IB	**CARD INIT**ialize. Input for starting the board (special case).
3	R_ERRIND	IB	**R**eset **ERR**or **IND**ication. Input to reset error indications.
4	IEVERR-N	IB	**I**ncremental **E**ncoder **V**oltage **ERR**or **- N**. Input for external signal for supervision of the pulse transmitter voltage. 0=voltage too low. 1=normal voltage.
5	SEL_IES	IB	**SEL**ect **I**ncremental **E**ncoder **S**upervision. Input for selection of internal or external supervision of the pulse transmitter. 0=via input IEVERR-N of the PC element. 1=via internal DI on DSDP 140A.
6	IE_TYPE	IB	**I**ncremental **E**ncoder **TYPE**. Input for selection of pulse transmitter type. 0=QGFA 110/120. 1=other types.
7	ENERRIND	IB	**EN**able **ERR**or **IND**ication. Input for selection of error indications via LED on DSDP 140A. 0=normal operation indication. 1=normal operation indication + error indication on LED.
8	L_VALUE	IR	**L**ength **VALUE**. Input for reference value for length measurement. Engineering unit.
9	START_LM	IB	**START L**ength **M**easurement. Input for starting condition for measurement and activation of the output.

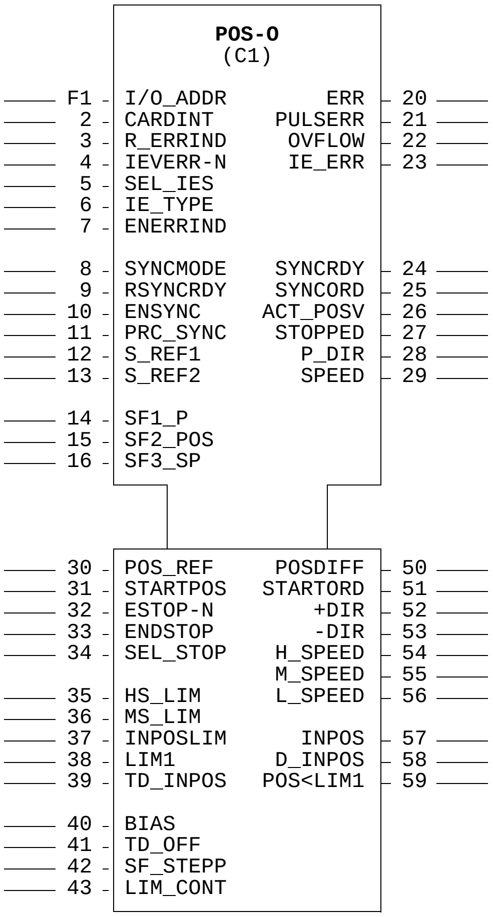
Terminal Description

No	Name	Type	Description
10	SEL_EXT	IB	**SEL**ect **EXT**ernal start order. Selection of external start order via DI1 or STROBE. 1=STROBE, leading edge (special transducer). 0=DI1
11	EXT_ST	IB	**EXT**ernal **START**. Input for selection of start conditions for length.
12	CONT_CON	IB	**CONT**inuous **CON**trol. Input for selection of continuous measurement and ordering. 0=new measurement must be started with START ORDER. 1=New measurement is started directly following COINC. The output signal ORDOUT is active for 200 ms. The counter is pre-loaded with the value at the input LVALUE.
13	SO_COINC	IB	**S**elect **S**top **O**n **COINC**idence. Input for selecting stop at the first coincidence. 0=Length measurement is stopped at the first coincidence. 1=Length measurement is restarted and counted to the length value of LVALUE on the condition that CONTCON=1.
14	LIM1	IR	**LIM**it **1**. Limit value 1 for length.
15	LIM2	IR	**LIM**it **2**. Limit value 2 for length.
16	SF1_P	II	**S**cale **F**actor **1** **P**ulse. Input for scale factor for pulses. The value at this input must be 1, 2 or 4. 1=count on the leading edge of channel A. 2=count on the leading edges of channels A and B.
17	SF2_L	IR	**S**cale **F**actor **2** **L**ength. Input for number of pulses/engineering unit. The value of SF1_P must be taken into account.
20	ERR	OB	**ERR**or. Output which, when set to 1, indicates that some error has occurred.
21	PULSERR	OB	**PULS**e **ERR**or. Output which, when set to 1, indicates an error in the pulse signal.
22	OVFLOW	OB	**OV**er**FLOW**/underflow. Output which, when set to 1, indicates an error in the counter for posvalue.
23	IE_ERR	OB	**IE ERR**or. Output which, when set to 1, indicates an error in the pulse transmitter, cable or signal.
24	M_ORD	OB	**M**easuring **ORD**er. Output set to 1 during measurement.
25	ORD_OUT	OB	**ORD**er **OUT**put. Output set to 1 at coincidence between LENGTH and LVALUE.
26	DIR_ERR	OB	**DIR**ection **ERR**or. Output for supervision of counting direction. 0=counting direction is downwards towards coincidence. 1=wrong counting direction
27	LENGTH	OR	Output for measured position value. The value starts at 0 and counts up to LVALUE, and further in certain cases.
28	L>LIM1	OB	**L**ength**>LIM1**. Indication that LENGTH>LIM1.
29	L>LIM2	OB	**L**ength**>LIM2**. Indication that LENGTH>LIM2.

Positioning, On-Off Control POS-O

Summary

POS-O (**POS**itioning **- O**n/off)is used with DSDP 140A for position control with digital output signal.



PC element POS-O

Call POS-O (C1)

Parameter	Significance	Permissible values
C1	Selection of control mode (internal area). C1=0 gives ON/OFF position control. C1=1 gives ON/OFF position control with push.	0 - 1

Terminal Description

No	Name	Type	Description
F1	I/O_ADDR	II	**I/O ADD**ress. Strapped I/O address on DSDP 140A in decimal code. The board must be strapped to an even address within the interval for Misc Board which is address 192 to 239.
2	CARDINIT	IB	**CARD INIT**ialize. Input for starting the board.
3	R_ERRIND	IB	**R**eset **ERR**or **IND**ication. Input for clearing an error indication.
4	IEVERR-N	IB	**I**ncremental **E**ncoder **V**oltage **ERR**or **- N**. Input for external signal for supervision of pulse transmitter voltage. 0=voltage too low. 1=normal voltage.
5	SEL_IES	IB	**SEL**ect **I**ncremental **E**ncoder **S**upervision. Input for selection of internal or external supervision of the pulse transmitter. 0=via input IEVERR-N of the PC element. 1=via internal DI on DSDP 140A.
6	IE_TYPE	IB	**I**ncremental **E**ncoder **TYPE**. Input for selection of pulse transmitter type. 0=QGFA 110/120. 1=other brands.
7	ENERRIND	IB	**EN**able **ERR**or **IND**ication. Input for selection of error indications via LED on DSDP 140A. 0=normal operation indication. 1=normal operation indication + error indication on LED.
8	SYNCMODE	II	**SYNC**hronization **MODE**. Input for selection of synchronization mode.
9	RSYNCRDY	IB	**R**eset **SYNCRDY**. Input for clearing of SYNCRDY. If this input is 0 the complete positioning system is blocked.
10	ENSYNC	IB	**EN**able **SYNC**hronization. Input which, when set to 1, permits synchronization according to condition SYNCMODE.
11	PRC_SYNC	IB	**PR**ogram **C**ontrolled **SYNC**. Pulse input for program controlled synchronization. This synchronization is performed independently of ENSYNC, and conditions according to SYNCMODE do not apply.
12	S_REF1	IR	**S**ynchronization **REF**erence **1**. Reference value in engineering unit for synchronization.Selected by signal SYNC 1.
13	S_REF2	IR	**S**ynchronization **REF**erence **2**. Reference value in engineering unit for synchronization.Selected with signal SYNC 2.
14	SF1_P	II	**SF1 P**ulse. Input for scale factor (2 bits). 1: A's leading edge. 2: A's and B's leading edges.4: A's and B's leading and trailing edges.
15	SF2_POS	IR	**SF2 POS**ition. Input for scale factor expressed as: Number of pulses/engineering unit. The value of SF1 selected must be considered when choosing the SF2 value. The PC element gives a reference scaled in pulses (S-REF*SF2) to the board.
16	SF3_SP	II	**SF3 SP**eed. Input for scale factor for speed.
20	ERR	OB	**ERR**or. Collective error signal.
21	PULSERR	OB	**PULS**e **ERR**or. Output which, when set to 1, indicates an error in the pulse signal.

Terminal Description

No	Name	Type	Description
22	OVFLOW	OB	**OV**er**FLOW**/underflow. Output which, when set to 1, indicates an error in the counter for pos value.
23	IE_ERR	OB	**IE ERR**or. Output which, when set to 1, indicates a fault in the pulse transmitter, cable or signal.
24	SYNCRDY	OB	**SYNC**hronization **R**ea**DY**. Output which, when set to 1, indicates that synchronization has been completed.
25	SYNCORD	OB	**SYNC**hronization **ORD**er. Output which, when set to 1, indicates that synchronization has been ordered. The output remains set for 5 seconds.
26	ACT_POSV	OR	**ACT**ual **POS**ition **V**alue. Output for position value in engineering unit.
27	STOPPED	OB	Output which, when set to 1, indicates that the object is stationary, that is, the position counter value has not changed since the previous reading.
28	P_DIR	OB	**P**ositive **DIR**ection. Output for indication of direction of movement. 1: Positive direction. 0: Negative direction.
29	SPEED	OI	Output, the value of which indicates the speed of the object expressed as number of pulses/90 ms, scaled by the factor SF3_SP.
30	POS_REF	IR	**POS**ition **REF**erence. Input for reference value for positioning.
31	STARTPOS	IB	**START POS**itioning. Input for starting conditions for positioning. The input is internally interlocked by means of the signals FAULT, SYNCRDY, ESTOP-N and DINPOS (if the object is in the correct position).
32	ESTOP-N	IB	**E**mergency **STOP-N**. Input which, when set to 0 interrupts positioning.
33	ENDSTOP	IB	**END STOP**. Input for external limit position. The input blocks the positioning.
34	SEL_STOP	IB	**SEL**ect **STOP** in position. 0: Continuous control of position according to the input POSREF. 1: Selection of stop via DINPOS, that is, the positioning is interrupted when the position corresponds to POSREF.
35	HS_LIM	IR	**H**igh **S**peed **LIM**it. Input for limit value for control at high speed.
36	MS_LIM	IR	**M**edium **S**peed **LIM**it. Input for limit value for control at medium speed.
37	INPOSLIM	IR	**IN POS**ition **LIM**it. Limit value for indication of correct position. The input indicates how much the real position may deviate from the reference and still be considered to be in position (INPOS set to 1).
38	LIM1	IR	**LIM**it **1**. Limit value for position.
39	TD_INPOS	II	**T**ime **D**elay **INPOS**. Input for time during which the object must be in the correct position (POSREF±INPOSLIM) before STARTORD is set to 0. Given in number of 20 ms intervals, that is, 0-255 x 20 ms.
40	BIAS	II	Input for addition to the pulse time calculated in the pos loop. Given in units of 5 ms.
41	TD_OFF	II	**T**ime **D**elay **OFF**. Input for pause time between pulses. Given in units of 5 ms.
42	SF_STEPP	II	**S**cale **F**actor **STEPP**. Input for factor for calculation of the rise time for a pulse.

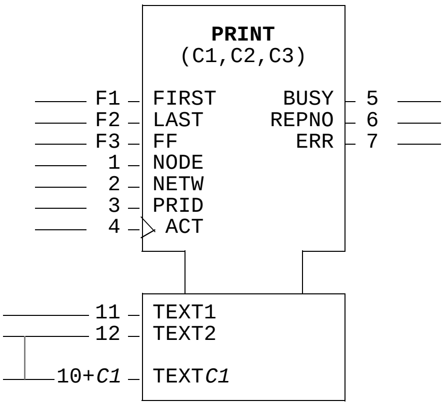
Terminal Description

No	Name	Type	Description
43	LIM_CONT	IR	**LIM**it **CONT**inuous area. Input for pos limit value for variable pulse width. If |POSDIFF|>LIMCONT no switching of the signal LSPEED is done.
50	POSDIFF	OR	**POS**ition **DIFF**erence. Output for control deviation, that is, POSREF - measured value (ACTPOSV).
51	STARTORD	OB	**START ORD**ered. Output for indication that positioning has been ordered via STARTPOS or FORWARD/REVERSE.
52	+DIR	OB	**+DIR**ection. Output which, when set to 1, indicates forward direction.
53	–DIR	OB	–**DIR**ection. Output which, when set to 1, indicates backward direction.
54	H_SPEED	OB	**H**igh**SPEED**. Output set at high speed.
55	M_SPEED	OB	**M**edium**SPEED**. Output set at high and medium speed.
56	L_SPEED	OB	**L**ow**SPEED**. Output set at high, medium and low speed. If |POSDIFF|> LIMCONT LSPEED is not switched for POS-O (1).
57	INPOS	OB	**IN POS**ition. Output that indicates that the set position has been reached, that is, |POSDIFF|<INPOSLIM.
58	D_INPOS	OB	**D**elayed **INPOS**. Output which, when set to 1, indicates that INPOS has applied for the duration TDINPOS. It is used to deactivate STARTORD.
59	POS<LIM1	OB	**POS**ition**<LIM1**. Indication that POS<POS LIM 1.

Print Request PRINT

Summary

PRINT is used to send one or more lines of text to a printer.



PC element PRINT

Call **PRINT (C1, C2, C3)**

Parameter	Significance	Permissible values
C1	Number of text lines to be sent to the printer	1 - 15
C2	Number of printout repetitions	1 - 90
C3	Number of characters on the TEXT inputs	1 - 80

Terminal Description

No	Name	Type	Description
F1	FIRST	FB	Set to 1 in the element which sends the first line of the printout.
F2	LAST	FB	Set to 1 in the element which sends the last line of the printout.
F3	FF	FB	**F**orm **F**eed. Set to 1 if form feed is required.
1	NODE	II	Number of the node where the required printer is located.
2	NETW	II	**NETW**ork. Number of the bus connecting the node with the PRINT element to the node with the printer.
3	PRID	II	**PR**inter **ID**entity. Number of the printer to be used. For MasterPiece, this number is always 1 and for MasterView 1 or 2.
4	ACT	IB	**ACT**ivate. Input for activation of printout. Printout starts in the cycle when the input goes to 1.
5	BUSY	OB	Set to 1 when another PRINT element in the same controller occupies the printer.
6	REPNO	OI	**REP**etition **N**umber. Specifies which repetition of the printout is currently in progress. The printout is repeated once per execution cycle and the number of repetitions is specified by call parameter C2.
7	ERR	OB	**ERR**or. Set to 1 when NODE or NETW can not be found or when PRID is wrong.

Terminal Description

No	Name	Type	Description
11	TEXT1	IA*C3*	Input for the first text line.
12	TEXT2	IA*C3*	Input for the second text line.
: 10+*C1*	TEXT*C1*	IA*C3*	Input for the last text line.

Function

The PC element PRINT is used to send a number of text lines to a printer. The printer and its control program can be located in the same system as the PRINT element or in another system which can be reached via a communication network. The element can send between 1 and 15 lines. Several PRINT elements can be used to generate longer printouts. Each line has a corresponding input TEXT1 - TEXTC1. If a blank line is required, the corresponding input is connected to an "empty" line.
In the first or only element which sends to the printer, the input FIRST is to be set to 1 to reserve the printer. FF, FIRST, and LAST govern the printers form feed function according to the table below.

Form Feed

FF	FIRST	LAST	Form feed
1	0	0	No
1	1	0	Before the printout
1	0	1	After the printout
1	1	1	Both before and after the printout
0	-	-	No

The input LAST is to be set to 1 in the last element of the printout. The FIRST and LAST inputs are to be reset to 0 in elements that are neither first nor last. The output BUSY indicates status for the element. When a request to send is detected BUSY is set to 1. When all text lines in the element have been sent to the printer BUSY is reset to 0.
REPNO indicates which repetition (1 - C2) is in progress.

Addressing of Printer

The inputs NODE, NETW and PRID determine which printer is to be used for the printout. NETW indicates which bus connects the node where PRINT is executed with the node where the printer is located. NODE indicates the number of the node where the printer is located. PRID specifies which printer in the specified node is to be used. If the node at which the printer is located is a MasterPiece system, this input is always to be 1 whereas that for a MasterView system can be 1 or 2. If the printer to be used for the printout is located in the local node, NODE and NETW are to be reset to 0.

Generation and Execution

All PRINT elements included in a list must be built under the same execution unit. Text lines may otherwise be lost. PRINT elements can only be executed under interpreter C.

Using Printer in Another Node

You can not use PC element outputs PRINT, ERR and BUSY to supervise printouts. Application must be made as follows:

Printer in MasterView 850/1

Controllers must not send their reports simultaneously. The printer used for reports must not be used for Alarms and Events (configure two printers in the MasterView 850/1).

Printer in Advant Station 500 Series

Controllers must not send their reports simultaneously.

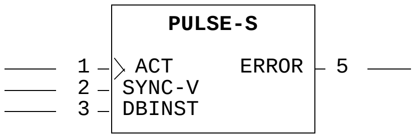
Several Controllers using printer in another Controller

Avoid sending from several Controllers simultaneously.

Synchronization of Pulse Counter PULSE-S

Summary

PULSE-S (**PULSE - S**ynchronize) is used to synchronize pulse counters on pulse counter boards, for example, DSDP 110.



PC element PULSE-S

Call PULSE-S

Terminal Description

No	Name	Type	Description
1	ACT	IB	**ACT**ivate. Input for activation of synchronization.
2	SYNC-V	IR	**SYNC**hronization **- V**alue. Input for synchronization value.
3	DBINST	II	**D**ata **B**ase **INST**ance. Input for instance number for the data base element PULSES.
5	ERROR	II	Output for error code, see the table below

Function

When the input ACT is set (to 1) the element attempts to synchronize a pulse counter on a pulse counter board. The pulse counter is identified by means of an instance number to be found in the header of the data base element PULSES. The instance number given at input DBINST is located in the header of the corresponding data base element. Synchronization is performed using the value at the input SYNC-V.

Supervision

The element has an ERROR output which presents the status of the element in accordance with the table below.

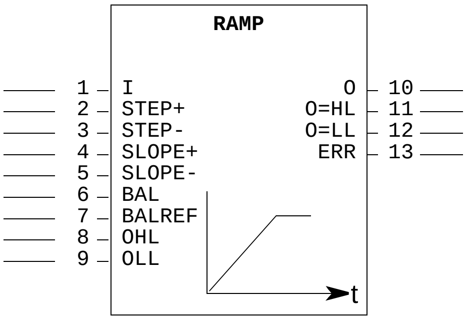
Interpretation of ERROR output values

Value	Meaning
–1	Synchronization not requested.
0	Synchronization done.
1	The value DBINST does not correspond to a pulse counter. The pulse counter is blocked. The value at input SYNC-V is unreasonable.
2	The value DBINST is illegal. Pulse counter or pulse counter board is not implemented.
3	Pulse counter being processed by the process communication. Repeat the attempt.
4	Pulse counter board has error indication.
5	Pulse counter board marked out-of-service.

Ramp Generator RAMP

Summary

RAMP (**RAMP** generator) is used to limit the speed of change of a signal. The output signal can be limited with limit values specified at special inputs. The balancing function permits the output signal to follow an external reference.



PC element RAMP

Call RAMP

Terminal Description

No	Name	Type	Description
1	I	IR	**I**nput. Input for the start value concerned.
2	STEP+	IR	The greatest allowed positive step change.
3	STEP–	IR	The greatest allowed negative step change.
4	SLOPE+	IR	Positive ramp for the output. Must be a positive number.
5	SLOPE–	IR	Negative ramp for the output. Must be a positive number.
6	BAL	IB	**BAL**ance. Input for activation of following.
7	BALREF	IR	**BAL**ance **REF**erence. Input for reference value when following.
8	OHL	IR	**O**utput **H**igh **L**imit. Input for upper limit value.
9	OLL	IR	**O**utput **L**ow **L**imit. Input for lower limit value.
10	O	OR	**O**utput. Output signal.
11	O=HL	OB	**O**utput=**H**igh **L**imit. Output which is set to 1 if the output O is limited to the upper limit value.
12	O=LL	OB	**O**utput=**L**ow **L**imit. Output which is set to 1 if the output O is limited to the lower limit value.
13	ERR	OB	**ERR**or. Output set to 1 if OHL is less than OLL.

Function

The main property of a RAMP function is that the output signal follows the input signal while the input signal is not changed more than the value specified at the step inputs. If the input signal is changed more than so, the output signal is first changed by STEP+ or STEP— depending on the direction of change and then by SLOPE+ or SLOPE— per second until the values at the input and output are equal. This means also that if STEP—=STEP+=0, a pure RAMP, that is, SLOPE/sec, is obtained at the output.

The greatest step change allowed at the output O is specified by the parameters STEP+ and STEP— for the respective direction of change. The ramp which the output signal is to follow if the change at the input I exceeds STEP+ or STEP— is specified by the inputs SLOPE+ and SLOPE—. All parameters are specified as absolute values with the same units as input I.

The values of the parameters are stored internally in the element. New values are only entered under stationary conditions, that is, when I(t)=O(t).

Certain constants are precalculated to make the execution time of the element as short as possible. The results are stored internally in the element. These constants are recalculated if SLOPE+ or SLOPE— are changed by more than 1/128 part of their previous values or if the sampling time TS is changed.

The second figure below shows the relationship between input, output and internal auxiliary variables when the element functions normally.

Calculation of Output Signal

When calculating output signal O, three cases must be distinguished:

- I(t)=O(t-TS) results in O(t)=I(t)
- I(t)>O(t-TS) results in calculation of the value of the output in accordance with:
- O(t)=min(I(t) and (Step++VPOS(t–TS)+SLOPE+ x TS), that is, the least of the values of I(t) and (Step++VPOS (t–TS)+SLOPE– x TS)
- I(t)<O(t–TS) results in calculation of the value of the output in accordance with: O(t)=max(I(t), VNEG (t-TS)–STEP—SLOPE–×TS.

VPOS and VNEG are auxiliary variables with positive and negative steps respectively. These are calculated in all cases according to:

- VPOS(t)=min(O(t), VPOS(t–TS)+SLOPE+ x TS)
- VNEG(t)=max(O(t), VNEG(t–TS)–SLOPE– x TS).

If a new step with the same direction of change appears at the input before the internal auxiliary signal has been updated the step part of the output signal is reduced. A step with the opposite derivative, however, takes full effect. This is because the auxiliary variable for the opposite direction always has been updated to O(t).

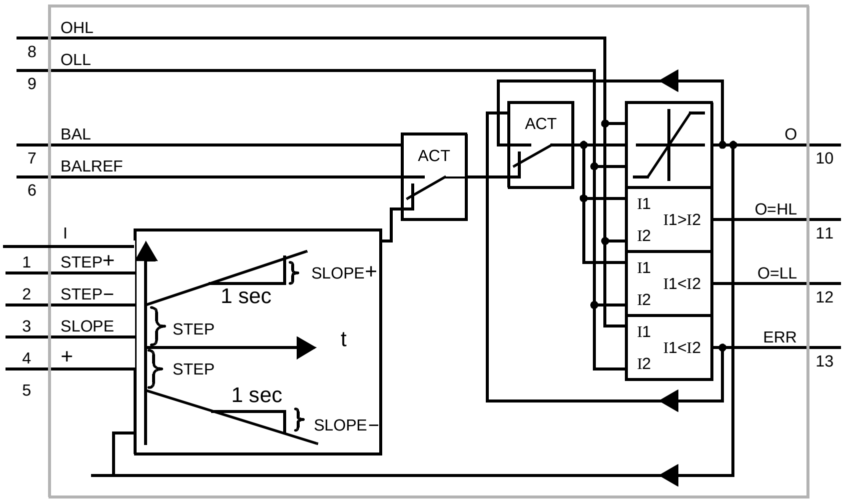
Following

If BAL is set to 1, the filter immediately goes into following and the output O is set to the value of the input BALREF.
If the value at BALREF exceeds the output signal limits, the output is set to the limit value concerned. During following VPOS(t)=VNEG(t)=O(t)=BALREF(t). Return to normal function is done as if a unit step had occurred on the input.

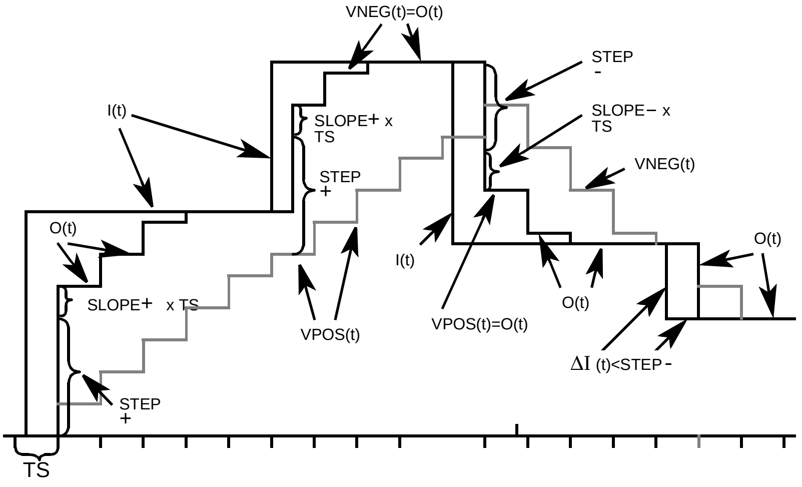
Limitation Function

The limitation function limits the output signal to the limit values at the inputs OHL for upper limit and OLL for the lower limit. If the actual value exceeds the upper limit, the output O=HL is set to 1 and if it falls below the lower limit, the output O=LL is set to 1. In the limiting state VPOS(t), VNEG(t) and O(t) are set to the limit value concerned.

The element checks that the upper limit value OHL is greater than the lower limit value OLL. If not, the output ERR is set to 1. While the error status persists, the outputs O=HL, O=LL and O retain the values they had in the sample before that in which the error occurred.



Function diagram

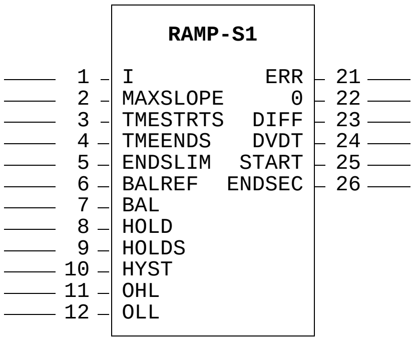


Relation between input, output and auxiliary variables under normal conditions

S-RAMP Generator RAMP-S1

Summary

RAMP-S1 (**S-RAMP** generator) generates a ramp signal function in which the transitions between constant outward signal and ramp are soft, that is with a continuous function.



PC element RAMP-S1

Call RAMP-S1

Terminal Description

No	Name	Type	Description
1	I	IR	**I**nput signal. Input for actual value.
2	MAXSLOPE	IR	**MAX**imum **SLOPE**. Ramp slope, max acceleration in m/s^2. If MAXSLOPE is set to zero, the condition for ending the ramp may be satisfied and the output signal "O" is momentarily set equal to the input signal "I".
3	TMESTRTS	ITR	**T**i**ME ST**a**RTS**. Time in seconds for the start-s phase of the ramp.
4	TMEENDS	ITR	**T**i**ME ENDS**. Time in seconds for the end-s phase of the ramp.
5	ENDSLIM	IR	**ENDS LIM**it. Limit that determines the characteristics of the s-ramp in the end-s phase to be connected from the output terminal DVDT.
6	BALREF	IR	**BAL**ance **REF**erence. Input for reference value on tracking.
7	BAL	IB	**BAL**ance. Input for activation of tracking BALREF.
8	HOLD	IB	**HOLD**. Instantaneous stop of the ramp.
9	HOLDS	IB	**HOLDS**. Orders hold with concluding end-s phase.
10	HYST	IR	**HYST**eresis. Hysteresis of the stationary state of the s-ramp.
11	OHL	IR	**O**utput **H**igh **L**imit. Input for upper limit value. OHL should not be changed dynamically.
12	OLL	IR	**O**utput **L**ow **L**imit. Input for lower limit value. OLL should not be changed dynamically.
21	ERR	OB	**ERR**or. Upward signal which is set to "true" if OHL£OLL.
22	O	OR	**O**utput signal. Output signal with s-form.
23	DIFF	OR	**DIFF**erence. Difference between input signal (I) and output signal (O).
24	DVDT	OR	**DVDT**. True first derivative. Expressed in current cycle time of the execution unit.

Terminal Description

No	Name	Type	Description
25	START	OB	**START**. Signal which is set to "true" when │DIFF│>│HYST│ that is the s-ramp function is activated.
26	ENDSEC	OB	**END SEC**tor. Signal which is set to "true" when the ramp is in the end-s phase.

Operation

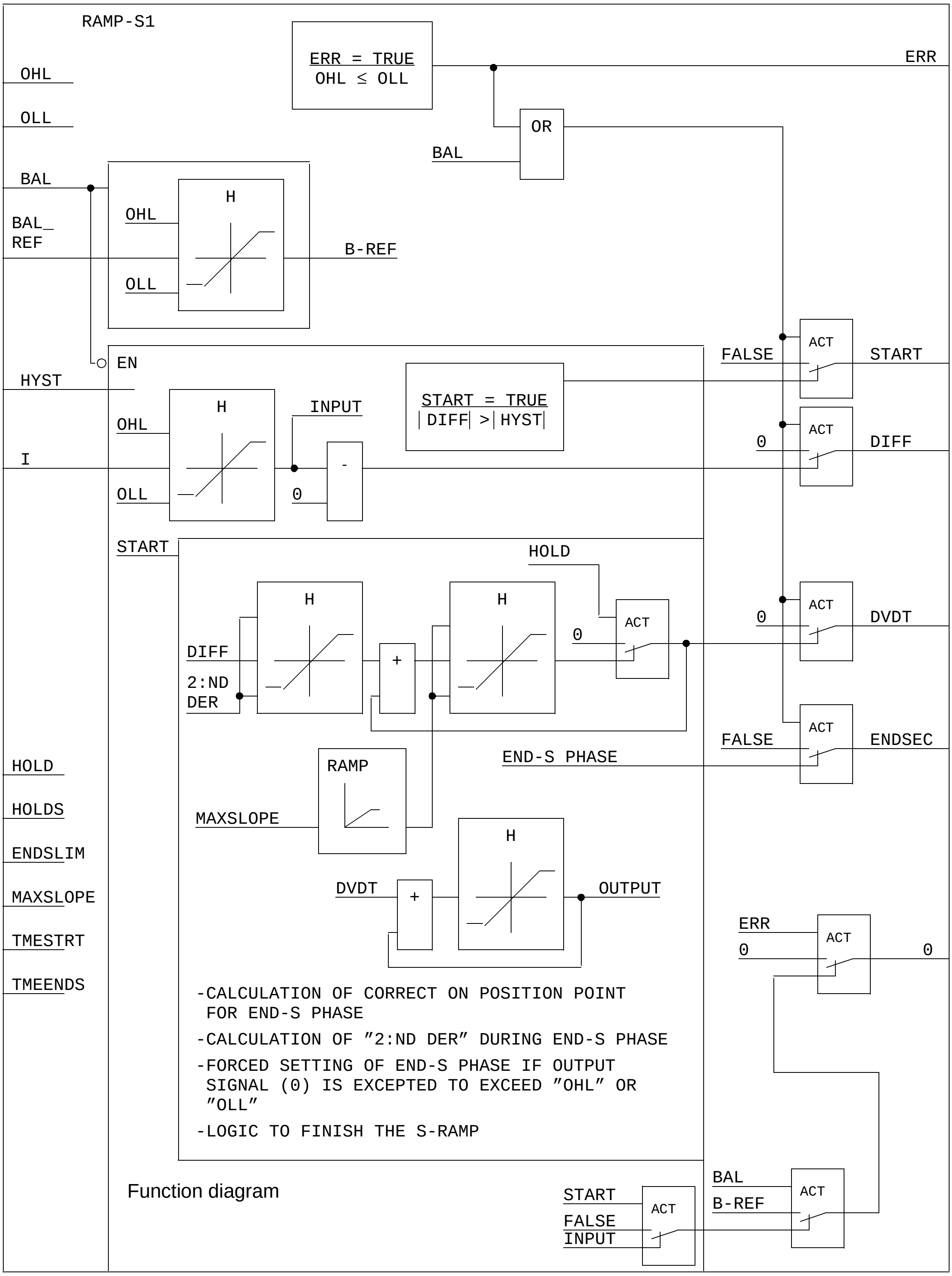
The s-ramp function is activated when abs (absolute) DIFF is greater than abs HYST. The second derivative is calculated from the parameter TMSTRTS and is constant except in the end-s phase. The calculated second derivative is integrated to a first derivative (DVDT) which is limited by the parameter MAXSLOPE. DVDT is added to the output signal (O).

When the s-ramp is activated, the element continuously calculates whether the end-s phase is to be switched in. During the end-s phase, the second derivative is calculated continuously to give a correct ending to the s-ramp. While the end-s phase is in progress, the s-ramp in unaffected by changes in MAXSLOPE and TMEENDS. When the s-ramp is activated, the element continuously checks whether the s-ramp can be ended, that is output signal (O) = input signal (I). The s-ramp can forcibly activate the end-s phase if the output (O) is calculated to go to the OLL or OHL limit; normal calculation of activation of the end-s phase is blocked.

HOLD gives an instantaneous stop of the ramp. HOLDS gives a rapid stop of the s-ramp with end-s phase. "ENDS" is used as the second derivative, provided that the s-ramp was in the end-s phase, in which case the end-s phase is completed.

When BAL is "true" BALREF is connected instantaneously to the output signal (O) without s-ramp function, DVDT and DIFF are reset and START and ENDSEC are set to "false". The HOLD and HOLDS function are blocked. When BAL becomes "false", the input signal (I) is connected via the s-ramp function to the output signal (O).

The input signals (I) and BALREF and the output signal (O) are limited by OLL and OHL. These should be constant. If they are to be changed, the order for the change must be interlocked so that the output signal (O) is not changed in a single step.

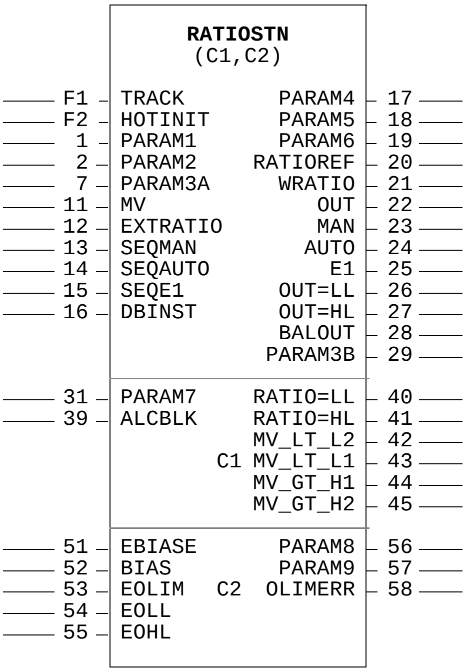


Function diagram

Ratio Station RATIOSTN

Summary

RATIOSTN forms part of the functional unit RATIOSTN (**RATIO ST**atio**N**). The unit is primarily used with the process controller PIDCON for ratio control. It is used in traditional instrumentation when a VDU based operator station is specified. Further information can be found in the manual *Functional Units part 4, PIDCON, RATIOSTN, MANSTN*.



PC element RATIOSTN

Call **RATIOSTN (C1, C2)**

The element consists of one part that forms a basic ratio station. This part may be extended with optional functions with the call parameters C1 and C2 according to the table below.

Parameter	Significance	Permissible values
C1	Selection of ratio limitation and measured value alarm.	0 - 1
C2	Selection of output signal limitation and external BIAS- setting	0 - 1

Terminal Description

No	Name	Type	Description
F1	TRACK	II	**TRACK**ing. Function parameter for tracking. 0=tracking. 1=RATIOREF follows EXTRATIO.
F2	HOTINIT	II	**HOT INIT**. Function parameter for warm start, see the HOTINIT table.
1	PARAM1	II	**PARAM**eter **1**. Connection to the data base element (PARAM1).

Terminal Description

No	Name	Type	Description
2	PARAM2	IA24	**PARAM**eter **2**. Connection to the data base element (PARAM2).
7	PARAM3A	II	**PARAM**eter **3A**. Connection to the data base element (PARAM3).
11	MV	IR	**M**easured **V**alue. Input for measured value.
12	EXTRATIO	IR	**EXT**ernal **RATIO**. Input for external ratio.
13	SEQMAN	IB	**SEQ**uential **MAN**ual. Pulse input for ordering of operation mode MAN.
14	SEQAUTO	IB	**SEQ**uential **AUTO**. Pulse input for ordering of operation mode AUTO.
15	SEQE1	IB	**SEQ**uential **E1**. Pulse input for ordering of operation mode E1.
16	DBINST	II	**D**ata **B**ase **INST**ance. An operational parameter or a constant, defining the instance number of the data base (to be found in its header). Alternatively, a connection to the data base element.
17	PARAM4	OI	**PARAM**eter **4**. Connection to the data base element (PARAM4).
18	PARAM5	OI	**PARAM**eter **5**. Connection to the data base element (PARAM5)
19	PARAM6	OA24	**PARAM**eter **6**. Connection to the data base element (PARAM6).
20	RATIOREF	OR	**RATIO REF**erence. Output for ratio reference with operation mode AUTO.
21	WRATIO	OR	**W**orking **RATIO**.
22	OUT	OR	**OUT**put. Output signal.
23	MAN	OB	**MAN**ual. Output for indication of operation mode MAN.
24	AUTO	OB	**AUTO**. Output for indication of operation mode AUTO.
25	E1	OB	Output for indication of operation mode E1.
26	OUT=LL	OB	**OUT**put=**L**ow **L**imit. Output for indication of output signal limiting. It is set to 1 when the output signal has been limited to the lower limit value.
27	OUT=HL	OB	**OUT**put=**H**igh **L**imit. Output for indication of output signal limiting. It is set to 1 when the output signal has been limited to the upper limit value.
28	BALOUT	OB	**BAL**ance **OUT**. Output for indication of operation mode MAN or AUTO.
29	PARAM3B	OI	**PARAM**eter **3B**. Connection to the data base element (PARAM3). Acknowledgement of switching of operation mode by the operator.
31(1)	PARAM7	IA32	**PARAM**eter **7**. Connection to the data base element (PARAM7).
39(1)	ALCBLK	IB	**AL**arm **C**ontrol **BL**oc**K**. Input for blocking of limit value alarm.
40(1)	RATIO=LL	OB	**RATIO**=**L**ow **L**imit. Output which is set to 1 when the ratio is limited by the lower limit.
41(1)	RATIO=HL	OB	**RATIO**=**H**igh **L**imit. Output which is set to 1 when the RATIO is limited by upper limit.
42(1)	MV_LT_L2	OB	**M**easured **V**alue_**L**ess **T**han_**L**ow **2**. Output which is set to 1 when the measured value falls below the alarm limit L2.
43(1)	MV_LT_L1	OB	**M**easured **V**alue_**L**ess **T**han_**L**ow **1**. Output which is set to 1 when the measured value falls below the alarm limit L1.

Terminal Description

No	Name	Type	Description
44(1)	MV_GT_H1	OB	**M**easured **V**alue_**G**reater **T**han_**H**igh **1**. Output which is set to 1 when the measured value exceeds the alarm limit H1.
45(1)	MV_GT_H2	OB	**M**easured **V**alue_**G**reater **T**han_**H**igh **2**. Output which is set to 1 when the measured value exceeds the alarm limit H2.
51(2)	EBIASE	IB	**E**xternal **BIAS E**nable. Input for selection of bias. 0=value from the operator’s keyboard 1=value from the external input BIAS.
52(2)	BIAS	IR	Input for BIAS.
53(2)	EOLIM	IB	**E**xternal **O**utput **LIM**it. Input for selection of external limit values for the output signal.
54(2)	EOLL	IR	**E**xternal **O**utput **L**ow **L**imit. External lower limit value for the output signal.
55(2)	EOHL	IR	**E**xternal **O**utput **H**igh **L**imit. External upper limit value for the output signal.
56(2)	PARAM8	OA8	**PARAM**eter **8**. Connection to the data base element (PARAM8).
57(2)	PARAM9	OR	**PARAM**eter **9**. Connection to the data base element (PARAM9)
58(2)	OLIMERR	OB	**O**utput **LIM**it **ERR**or. Output which is set to 1 when EOLL>EOHL.

(1) These connections are available only if the call parameter C1 is set to 1.
(2) These connections are available only if the call parameter C2 is set to 1.

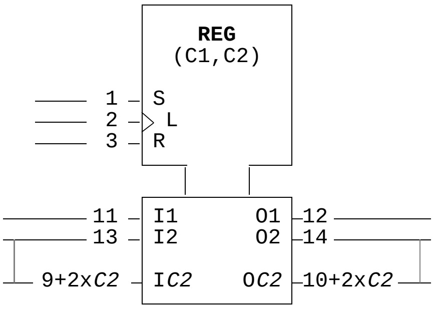
HOTINIT

	Function with warm start	
HOTINIT	**Operation mode**	**Output signal**
0	unchanged	unchanged
1	MAN	unchanged
2	MAN	0 or OLL
By unchanged is meant no change in relation to the state in effect before the event that caused the warm start.		

Register REG

Summary

Register REG (**REG**ister) is used as a memory element with an optional number of positions. The value of the data can be integer, real number, Boolean or time.



PC element REG

Call REG (C1, C2)

Parameter	Significance	Permissible values
C1	Data type	I, IL, R, B, T, TR
C2	Number of places	1 - 35

Terminal Description

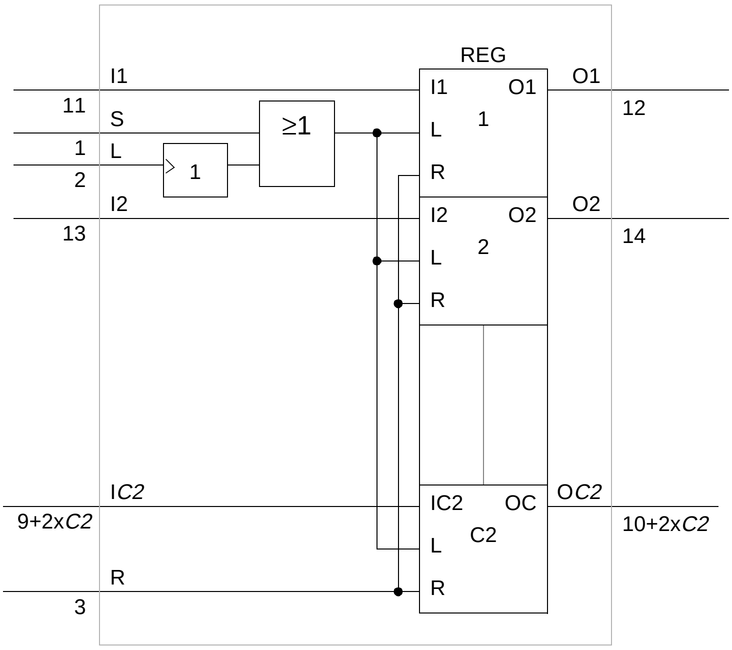
No	Name	Type	Description
1	S	IB	Set. Input for loading new data, each time the element is executed. When S is set to 0, the last data loaded remains.
2	L	IB	**L**oad. Dynamic input for loading data to the register. Loading is performed when L goes to 1.
3	R	IB	**R**eset. Input for clearing the register. R prevents all further entry while it is 1.
11	I1	I*C1*	**I**nput **1**. Input data to position 1.
12	O1	O*C1*	**O**utput **1**. Output data from position 1.
13	I2	I*C1*	**I**nput **2**. Input data to position 2.
14	O2	O*C1*	**O**utput **1**. Output data from position 2.
: 9+2x*C2*	I*C2*	I*C1*	**I**nput ***C2***. Input data to position C2.
10+2x*C2*	O*C2*	O*C1*	**O**utput ***C2***. Output data from position C2.

Function

When input L becomes 1, the register is loaded with data from inputs I1 - I*C2*. Data previously in the register is replaced.
If input S is set, loading is performed as above each time the register is executed. When S is reset after having been set, the data most recently loaded remains until the element is executed again, with input S, L or R set. Input S overrides L so that when input S is set, L has no effect.

Clearing

When input R is set, the register is cleared and all further entry is prevented. R overrides both S and L.

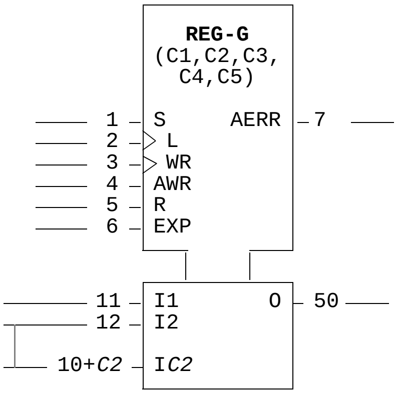


Function diagram

Group Data Register REG-G

Summary

REG-G (**REG**ister **- Gr**oup data) is used to assemble individual variables to a single variable of group type. The data type can be integer, real number, Boolean or time.



PC element REG-G

Call REG-G (C1, C2, C3, C4, C5)

Parameter	Significance	Permissible values
C1	Data type	I, IL, R, B, T, TR
C2	Number of inputs	0 - 30
C3	Selection of expander input EXP	0 - 1
C4	Number of places in group data for expander input EXP	0 - 254
C5	Number of places in group data for the output O.	0 - 255
C2+C3 x C4=C5		

Terminal Description

No	Name	Type	Description
1	S	IB	**S**et. Input for loading new data each time the element is executed. The latest loaded data remains when S is reset.
2	L	IB	**L**oad. Dynamic input for loading data to the register. Loading is performed when L goes to 1.
3	WR	IB	**WR**ite. Dynamic input for changing of data at the place specified by AWR.
4	AWR	II	**A**ddress **WR**ite. Input for address to the data to be changed according to WR.
5	R	IB	**R**eset. Input for clearing the register. R prevents all further entry while at 1.
6	EXP	IG*C4C1*	**EXP**ander input. Input for group data to be linked with data at inputs I1 - I*C2*.
7	AERR	OB	**A**ddress **ERR**or. Output which is set (to 1) if the address at AWR is greater than the number of inputs or is negative.

Terminal Description

No	Name	Type	Description
11	I1	I*C1*	**I**nput **1**. Input data to position 1.
12	I2	I*C1*	**I**nput **2**. Input data to position 2.
: 10+*C2*	I*C2*	I*C1*	**I**nput ***C2***. Input data to position C2.
50	O	OG*C5C1*	**O**utput. Output for group data.

Function

REG-G combines individual variables of the same data type to a single group variable. The element has an expander input EXP for a group variable which can be used when more than 30 variables are to be linked together. Input S overrides L so that when input S is set, L has no effect.

Normal Assembly

When input S is set, data is continuously assembled at the group variable of the output. The group variable of the output consists firstly of group data from the input EXP and then of the values from the inputs I1 - I*C2*. The element acts as a latch when the input S is reset, the latest data assembled then remaining at the output.

Loading

If input L goes from 0 to 1, an assembly is performed to output O during this program cycle in the same way as during normal assembly. If S or R is set, L has no effect.

Changing of Data at an Optional Place

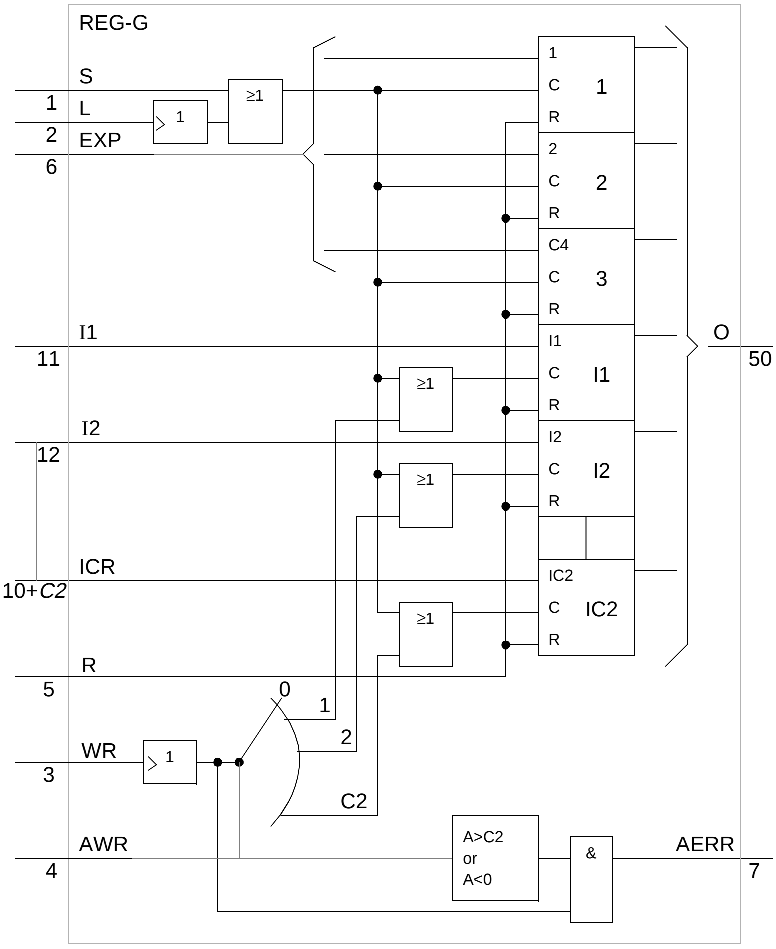
One of the ordinary data inputs I1 - I*C2* is used for changing of data at a special place in REG-G. The address (integer 1 - C2) of the place at which data is to be changed is stated at the input AWR. The new data value is entered from the input to the relevant place at the moment that the input WR goes to 1. If AWR = 0 and the input WR goes to 1, group data is read from the input EXP to their different places. Places which correspond to ordinary inputs are not affected.

Clearing

When input R is set, data at all places in the group register are cleared and all further entry is prevented. R overrides both S and L.

Supervision

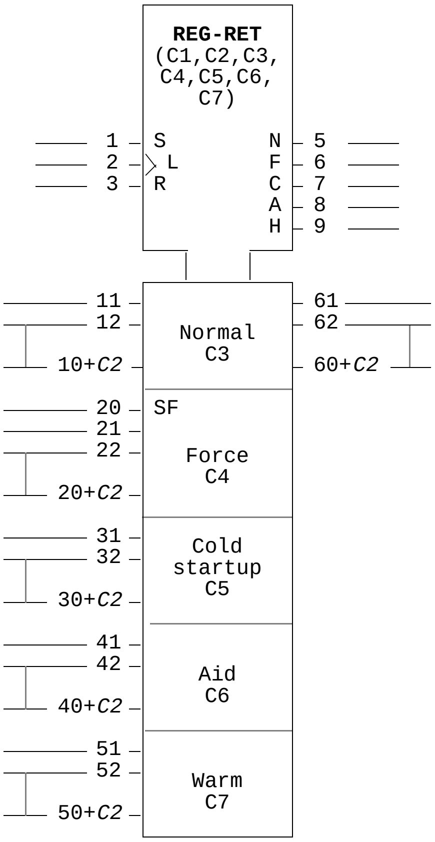
The address at the input AWR is checked and if its value is greater than the number of inputs or negative, the error signal output AERR is set. When an error is detected AERR is set during one cycle if input WR is set. No place in the register is affected when an error occurs.



Function diagram

Register with Retention REG-RET

Summary



PC element REG-RET

REG-RET (**REG**ister **-** with **RET**ention) is used as a register when clearing of data is not desired when the PC program changes state. Default values may also be defined for the different states. The data type and the number of channels (1 - 9) may be selected with a call parameter. The register has five groups of inputs that may be independently selected with call parameters. These groups are used to update the outputs depending on the state of the PC program concerned.

Call REG-RET (C1, C2, C3, C4, C5, C6, C7)

Parameter	Significance	Permissible values
C1	Data type	I, IL, R, B, T, TR
C2	Number of channels	1 - 9
C3	Normal function	0 - 1
C4	Force set function	0 - 1
C5	Cold start function	0 - 1
C6	Aid function	0 - 1
C7	Warm start function	0 - 1

Terminal Description

No	Name	Type	Description
1	S	IB	**S**et. Input for loading new data each time the element is executed. Data is fetched from the inputs 11 - 10+C2. S overrides L.
2	L	IB	**L**oad. Dynamic input for loading data. Loading is performed when L goes to 1.
3	R	IB	**R**eset. Input for clearing of the register. R overrides SF, S and L.
5(1)	N	OB	**N**ormal. Output that indicates that the outputs have been updated with normal values, that is, updated by setting S to 1 or by dynamic loading using L. N is set to 0 when R is set to 1.
6(2)	F	OB	**F**orce. Output that indicates that the outputs have been updated with force set values, that is, by setting SF to 1.
7(3)	C	OB	**C**old. Output that indicates that the outputs have been updated with cold start values when the command DIBM (**DI**sable **B**uild **M**ode) was given.
8(4)	A	OB	**A**id. Output that indicates that the outputs have been updated with aid values, that is, when the command EBM (**E**nable **B**uild **M**ode) or BLRS (**BL**ock and **R**e**S**et) was given at the aid. Updating with aid values is also performed when the PC program or execution units are reset.
9(5)	H	OB	**H**ot. Output that indicates that the outputs have been updated with warm start values, that is, after a warm start or the aid command DICONFIG.
11	-	I*C1*	Input 1 for normal data.
12	-	I*C1*	Input 2 for normal data.
: 10+*C2*	-	I*C1*	Input C2 for normal data.
20(2)	SF	IB	**S**et **F**orce. Input for loading new data each time the element is executed. Data is taken from the inputs 21 - 20+C2. SF overrides the inputs S and L.
21(2)	-	I*C1*	Input 1 for force set data.
22(2)	-	I*C1*	Input 2 for force set data.
: 20+*C2*(2)	-	I*C1*	Input C2 for force set data.
31(3)	-	I*C1*	Input 1 for cold start data.
32(3)	-	I*C1*	Input 2 for cold start data.
: 30+*C2*(3)	-	I*C1*	Input C2 for cold start data.
41(4)	-	I*C1*	Input 1 for aid data.
42(4)	-	I*C1*	Input 2 for aid data.
: 40+C2(4)	-	I*C1*	Input C2 for aid data.
51(5)	-	I*C1*	Input 1 for warm start data.
52(5)	-	I*C1*	Input 2 for warm start data.

Terminal Description

No	Name	Type	Description
: 50+*C2*[5]	-	I*C1*	Input C2 for warm start data.
61	-	O*C1*	Output 1 for data.
62	-	O*C1*	Output 2 for data.
: 60+*C2*	-	O*C1*	Output C2 for data.

(1) The output N exists only if the call parameter C3=1
(2) These connections only exist if the call parameter C4=1
(3) These connections only exist if the call parameter C5=1
(4) These connections only exist if the call parameter C6=1
(5) These connections only exist if the call parameter C7=1

Function

If the input S is set when the element is executed the outputs (61 - 60+*C2*) will be updated with normal values, that is, from the inputs 11 - 10+*C2*. If the call parameter C3 is set to 1, the output N will indicate when the outputs of the element have been updated with normal values by setting S or L to 1. This indication remains until R (**R**eset) is set to 1. The input R (**R**eset) clears all outputs. R overrides FS, S and L. If the call parameter C4 is set to 1, functions and inputs and outputs for force setting of data are included.

When force setting, the outputs 61 - 60+*C2* are updated with the values at the inputs 21 - 20+*C2*. The output F is set to 1 to indicate that the outputs have been force set. Force setting is performed by setting the input SF (20) to 1. The input SF overrides S and L.

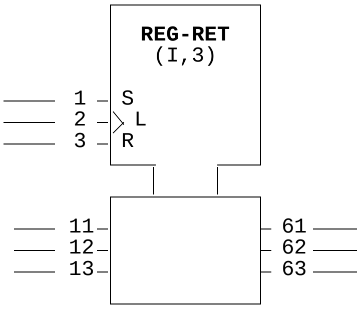
Cold start functions are selected by setting the call parameter C5 to 1. The cold start values are used when the command DIBM is given from a connected aid. The values at the inputs 31 - 30+*C2* are stored at the outputs 61 - 60+*C2* and the output C is set to 1.

Aid functions are selected by setting the call parameter C6 to 1. The outputs of the element are normally not affected by the Reset input of a superior PC program or execution unit. If the aid function is selected the outputs will be set to the aid values in case of a superior reset. When the command EBM or BLRS is given from a connected aid, the values at the inputs 41 - 40+C2 are stored at the outputs 61 - 60+*C2* and the output A is set to 1. If these outputs are connected to the data base, the values in the data base will remain zero as long as the element is in build mode.

Warm start functions are selected by setting the call parameter C7 to 1. When a warm start is performed the values at the inputs 51 - 50+*C2* are stored at the outputs 61 - 60+*C2* and the output H is set to 1. Restart after a power failure corresponds to a warm start. When any one of the outputs N, F, C, A or H is set to 1 as described above the other ones will be cleared.

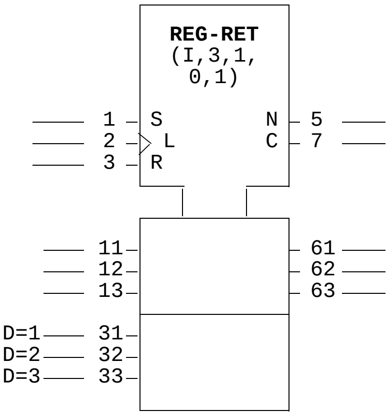
Example 1

REG-RET is used to store 3 integers (I) that must not be affected by any changes in the state of the program, see the figure below.



REG-RET (I,3)

Since C6 is 0 the output signals will not be affected by the command DIBM. As a consequence the outputs 61 - 63 do not have defined values until the element has been executed with S or R set to 1 or L has changed from 0 to 1. If it is required that the outputs have defined values prior to the first normal execution of the element, the function for cold start data must be selected, see the figure below.

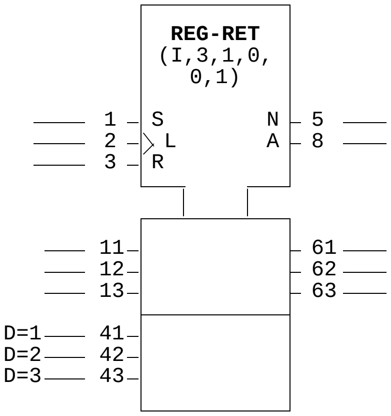


REG-RET (I,3,1,0,1)

The outputs N and C show whether or not the outputs 61 - 63 have updated values. After the command DIBM the outputs 61 - 63 will assume the values 1, 2 and 3 respectively, the output N is set to 0 and C to 1. These values will remain until the outputs are updated by execution with S set to 1 or by L becoming 1 after having been 0. During this execution the outputs 61 - 63 will receive their values from the inputs 11 - 13, N will be set to 1 and C to 0. If the input R is activated all outputs will be cleared.

Example 2

The element is to store values that are not to be affected by changes in program state, warm start or by the command DICONFIG. When the element is transferred to build mode, the outputs are to assume predetermined values, see the figure below.

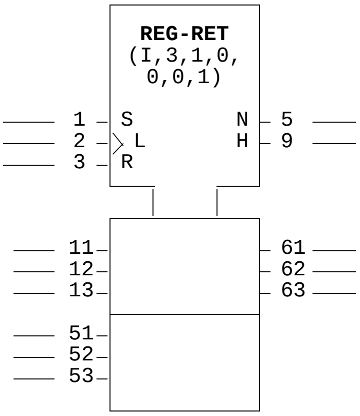


REG-RET (I,3,1,0,0,1)

The outputs of the element will, after the command EBM, assume the values 1, 2 and 3 respectively. Since the data base is always cleared when the command EBM is given, these values will not be stored in the data base until the command DIBM is given. The values at the outputs will remain until the element is executed with S set to 1 or L becoming 1 after having been 0, at which point the values at the inputs 11 - 13 are transferred to the outputs.

Example 3

At warm start the outputs are to be updated with predetermined values (at least one of which is not 0), see the figure below. At power-up or when the command DICONFIG is given, the outputs will assume the values given at the inputs 51 - 53, that is, 1, 2 and 3 in the figure. These values remain until the outputs are updated with the S or L input.

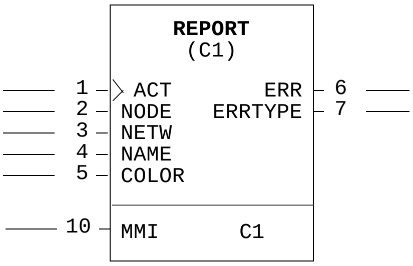


REG-RET (I,3,1,0,0,0,1)

Report Printout REPORT

Summary

REPORT is used to send a request to an operator station (Advant Station 500 Series or MasterView 800) for a printout on a hardcopy unit of a report display. The name of the report display is given as input data to the element. Which operator station to use and which MMI connected it is also given as input data to the element.



PC element REPORT

Call REPORT (C1)

Parameter	Significance	Permissible values
C1	Use MMI number selection for MasterView 800.	0 - 1 Default 0

Terminal Description

No	Name	Type	Description
1	ACT	IB	**ACT**ivate. Dynamic input for activation of the element.
2	NODE	II	Input for **NODE** number of the operator station where the report is to be presented.
3	NETW	II	**NETW**ork. Net number of the bus connecting the process station with the operator station where the report is to be presented.
4	NAME	IA20	Input for the name of the display to be presented.
5	COLOR	II	**MasterView:** Input for color code for the hardcopy unit. –1 No hardcopy printout. 0 Color code from an external computer. 1 Black/white 2 Black/white/red 3 7 colors. **AS 500 operator station:** –1: The display will be presented as a base on the defined screen. –2 : The display will be presented as an overlap display on the defined screen.
6	ERR	OB	**ERR**or. Output which indicates an error in the connection to the MasterView unit.
7	ERRTYPE	OI	**ERR**or **TYPE**. Output for error code (when ERR is set to 1). 0: Transmission OK. 1: No connection via channel to the operator station. 2: "Busy"; the channel is busy. The element makes three attempts to send the message, after which further attempts are aborted. 4: A message has after three attempts at transmission still not been received at the operator station.

Terminal Description

No	Name	Type	Description	
10	MMI	II	**MasterView:** **MMI** number within the MasterView 800 where the report is to be presented. Permissible values are: 0 - 3 for MasterView 850 0 - 2 for MasterView 830. 0 - 1 for MasterView 810 A value of 0 (or if C1 is 0) means that the MasterView 800 will use a default MMI number. Negative values or values >=4 will be set to 0 by the element.	**AS 500 operator station:** 1. The display will be presented on screen 1. 2. The display will be presented on screen 2. 3. Postscript generated copies will be printed without presenting the display on the screen The property color has no meaning in this case.

Function

The element gathers data from inputs NAME (the name of the report display) and COLOR (color code for hardcopy unit). The data is sent, when the input ACT is toggled from 0 to 1, as a request for a report printout or a display exchange to the operator station given by the inputs NETW and NODE. Which MMI unit to use within the operator station is given by the input MMI.

Error Handling

The output ERR is set to 1 if an error occurs. The output ERRTYPE indicates the error type. See output ERRTYPE in the table above. In normal operation ERR is clear and ERRTYPE=0.

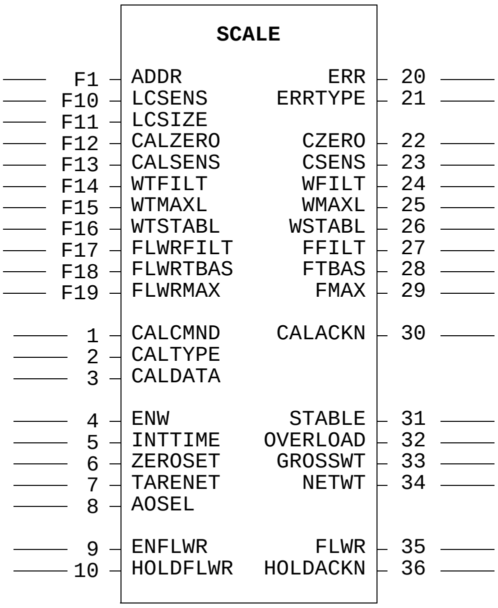
Weighing

SCALE

Summary

The SCALE element is used in conjunction with a weighing board, for example, DSXW 110/ 111, for weighing in Advant Controller 410/450. Typical applications are container and platform weighing. SCALE performs the logical interfacing between PC programs and the local weighing function on the weighing board. One SCALE element is needed for every scale to be included. For batching, see the SCALEDOS element. For a detailed description of the weighing function, see the manual Weighing, User's Manual.

Call SCALE



PC element SCALE

Terminal Description

No	Name	Type	Description
F1	ADDR	II	**ADDR**ess. Strapped I/O address on DSXW 110/111 in decimal code. The board must be strapped within the interval for Misc Board which is address 192 to 239.
F10	LCSENS	IR	**L**oad **C**ell **SENS**itivity. Sensitivity of the load cell in mV/V. Typically 1-3 mV/V.
F11	LCSIZE	IR	**L**oad **C**ell **SIZE**. The sum of the nominal capacity of the load cells, for example, 3×100 kg=300 kg for a container scale with three 100 kg load cells. The value is specified in the unit (kg, ton, lbs) the gross/net weights have to have.
F12	CALZERO	IR	**CAL**ibration **ZERO** level. Zero level after calibration. Normally not set. Default value=0. **NOTE:** The value is read from the weighing board and stored at the input.
F13	CALSENS	IR	**CAL**ibration **SENS**itivity. Sensitivity after calibration. Normally not set. Default value=0. **NOTE:** The value is read from the weighing board and stored at the input.
F14	WTFILT	IR	**W**eigh**T FILT**er constant. Filter constant for filtration of weight value. Value is given in seconds.
F15	WTMAXL	IR	**W**eigh**T MAX**imum **L**imit. Limit of maximum load on scale (overload limit). The value of the gross weight GROSSWT and/or the net weight NETWT that corresponds to +10 V on the analog output.
F16	WTSTABL	IR	**W**eigh**T STAB**le **L**imit. Limit of stable weight.

Terminal Description

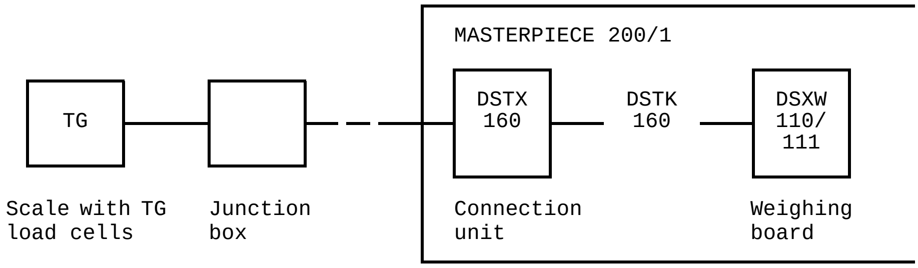
No	Name	Type	Description
F17	FLWRFILT	IR	**FL**o**WR**ate **FILT**er constant. Filter constant for filtering weight flow rate. Value in seconds.
F18	FLWRTBAS	II	**FL**o**WR**ate **T**ime **BAS**e. Time base to calculate weight flow rate. **Selection** **Time base** 1 Second 2 Minute 3 Hour
F19	FLWRMAX	IR	**FL**o**WRate. MAX**imum value. The value of the weight flow FLWR that corresponds to +10 V on the analog output.
1	CALCMND	IB	**CAL**ibration **C**o**M**ma**ND**. Activates calibration according to selected type.
2	CALTYPE	II	**CAL**ibration **TYPE**. Type of calibration. **Type** **Method** 1 Determine zero level 2 Calibration with load cell size 3 Calibration with load cell sensitivity 4 Calibration with known weight 5 Correction of calibration
3	CALDATA	IR	**CAL**ibration **DATA**. Data associated with type of calibration. Type Data 1 No meaning 2 Load cell size given in the unit to be used for the weighing (kg, ton, lbs) 3 Load cell sensitivity in mV/V 4 Known weight given in the unit to be used for the weighing (kg, ton, lbs) 5 Correcting factor. Should be given in the same quantity as you want the weight to be given in.
4	ENW	IB	**EN**able **W**eighing. Activates weight calculation and enables error reporting via the ERR and ERRTYPE outputs.
5	INTTIME	IR	**INT**egration **TIME**. Integration time for A/D conversion. To be selected between 0.01,..., 0.1 s.
6	ZEROSET	IB	**ZEROSET**. Sets gross weight to zero. **NOTE:** The tare weight for the net weight calculation is set to zero, and therefore the net weight is also set to zero.
7	TARENET	IB	**TARE NET**. Tares the net weight, that is, the net tare weight is set equal to the actual gross weight. The net weight is set to zero.
8	AOSEL	II	**A**nalog **O**utput **SEL**ect. Selection of value for analog output. **Selection** **Output 1** **Output 2** 1 Gross weight Net weight 2 Gross weight Weight flow 3 Net weight Weight flow
9	ENFLWR	IB	**EN**able **FL**o**WR**ate. Activates calculation of weight flow rate.
10	HOLDFLWR	IB	**HOLD FL**o**WR**ate. Freezes calculation of weight flow rate.
20	ERR	OB	**ERR**or. Indicates error.
21	ERRTYPE	OI	**ERR**or **TYPE**. Indicates type of error.

Terminal Description

No	Name	Type	Description
22	CZERO	OR	**C**alibration **ZERO** level. Copy of CALZERO.
23	CSENS	OR	**C**alibration **SENS**itivity. Copy of CALSENS.
24	WFILT	OR	**W**eight **FILT**er constant. Copy of WTFILT.
25	WMAXL	OR	**W**eight **MAX**imum **L**imit. Copy of WTMAXL.
26	WSTABL	OR	**W**eight **STAB**le **L**imit. Copy of WTSTABL.
27	FFILT	OR	**F**lowrate **FILT**er constant. Copy of FLWRFILT.
28	FTBAS	OI	**F**lowrate **T**ime **BAS**e. Copy of FLWRTBAS.
29	FMAX	OR	**F**lowrate **MAX**imum value. A copy of FLWRMAX.
30	CALACKN	OB	**CAL**ibration **ACKN**owledge. Acknowledgement of completed calibration.
31	STABLE	OB	**STABLE**. Indicates stable scale, that is, the weight change between two measured values is below the limit for stable rate (ΔWT<WTSTABL).
32	OVERLOAD	OB	**OVERLOAD**. Indicates scale overload, that is, gross weight exceeds limit for maximum permitted load (GROSSWT>WTMAXL).
33	GROSSWT	OR	**GROSS W**eigh**T**. Gross weight.
34	NETWT	OR	**NET W**eigh**T**. Net weight, that is, gross weight minus net tare weight.
35	FLWR	OR	**FL**o**WR**ate. Weight flowrate, that is, weight change per unit of time (second, minute, hour)
36	HOLDACKN	OB	**HOLD ACKN**owledge. Acknowledgement of freezing of calculation of weight flow rate.

Function, General

The SCALE element is used to connect weighing data/signals produced on a weighing board, for example, DSXW 110/ 111, to PC programs. Every scale to be included requires one SCALE element with the associated weighing board, connection cable, connection unit, junction box load cells and any other necessary cabling.



Supplementary function Weighing, block diagram

The most important data produced is gross weight (GROSSWT), net weight (NETWT) and weight flow rate (FLWR). These data are calculated on the weighing board and SCALE mirrors them at the GROSSWT, NETWT and FLWR outputs. A number of function parameters and variables are associated with them. The filters for weight and weight flow rate are realized in software on the weighing board and are in the nature of two series connected RC links, that is, they both have the transfer function $F(s)=1/(1+sT)^2$, where T is equal to WTFILT or FLWRFILT in seconds. The weight flow rate is calculated from unfiltered weight values.

The gross weight GROSSWT and the net weight NETWT are both given in the units specified at the time of calibration (kg, ton, lbs). The rate of flow, FLWR, receives its dimension (kg/s, kg/m, kg/h, ton/s, ton/m, ton/h, lbs/s, lbs/m, lbs/h) from the weight unit specified at a calibration combined with the time base specified by the function parameter FLWRTBAS.

The outputs CZERO, CSENS, WFILT, WMAXL, WSTABL, FFILT, FTBAS and FMAX are included to mirror the corresponding function parameters CALZERO, CALSENS, WTFILT, WTMAXL, WTSTABL, FLWRFILT, FLWRTBAS and FLWRMAX and thus permit linking via PC programs to object display in Advant Station 500 OS.

Calibration

Calibration is the setting of the zero point value (CALZERO) and sensitivity (CALSENS) of the scale. These two values are calculated locally on the weighing board during the calibration procedure.After calibration, the values of CALZERO and CALSENS are transferred to the SCALE element, and written to the functions parameter inputs with the same name. In this way these values are held in the PC program and are included in source code dumps. Normally, therefore, these inputs need not be defined when using SCALE, since they are both given the initial value 0. There are cases when they must be defined, but for this the values must have previously been read off after a correct calibration.

The scale is calibrated by manipulating the inputs CALCMND, CALTYPE, CALDATA and the output CALACKN. CALCMND must be set to 1 to start a calibration procedure. CALACKN acknowledges completed calibration. CALTYPE defines the type of calibration to be performed. Initially, Type 2 is often used, that is, calibration with load cell size, and this gives a reasonable and sometimes sufficient level of accuracy. For Type 4, an accurately known weight must be available. This weight is placed on the scale during calibration, and the sensitivity of the scale is calculated on the basis of the specified weight value. There is also Type 5, an option for correcting an existing calibration.

Error Codes

The outputs ERR and ERRTYPE specify the error states that can occur. ERR is a logic output that is set to 1 when an error is detected in the transmission of data from/ to the weighing board. The output ERRTYPE mirrors the error state with a code as shown in the table below.

Error	Error Description Code
–2	Calibration data absent, scale not calibrated
–1	Communication with weighing board not initiated
0	No error
1	Incorrect ’BOARD TYPE’ specified in data base
2	Bus malfunction
3	Fault in communication with weighing board
4	Error in initiation of weighing board
5	Weighing board missing
6	Data fault, SCALEDOS (BINSEL)
7	Data fault, SCALEDOS (LOADSEL)
11	Hardware fault, ±15 V stabilizer
12	Hardware fault, load cell supply
13	Hardware fault, A/D converter measurement references
14	Hardware fault, A/D converter test references
15	Hardware fault, A/D converter counter
16	Hardware fault, ”time out” for A/D conversion
17	Measured value error, overflow in measurement channel offset
18	Measured value error, overflow in test reference
19	Measured value error, overflow in reference quote
20	Measured value error, overflow in weight value
21	Address fault on PC element

Batching

SCALEDOS

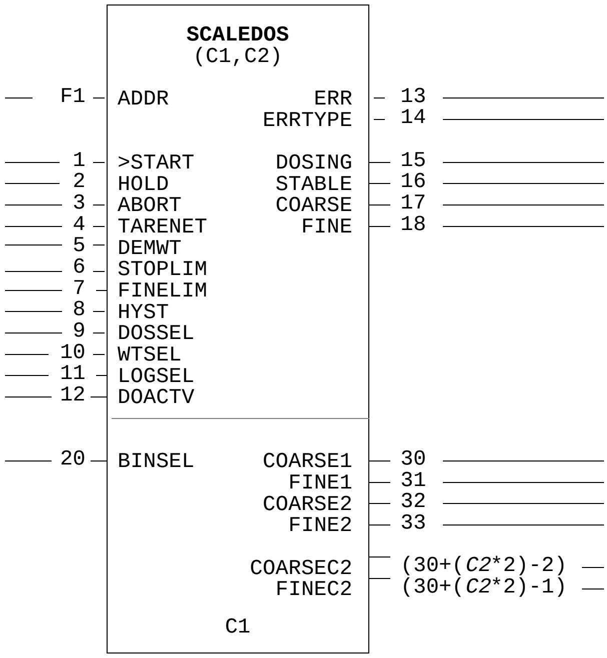
Summary

The SCALEDOS element is used together with a weighing card, for example, DSXW 110/ 111, for batching in Advant Controller 410/450. SCALEDOS provides the logical interfacing between PC programs and the local batching function on the weighing board. One SCALEDOS element is needed for every batching scale included in the system.

NOTE

> For SCALEDOS to be used, the SCALE element must be used in PC programs for the actual weighing function.

For a detailed description of the dosage function, see the manual Weighing User's Manual.



PC element SCALEDOS

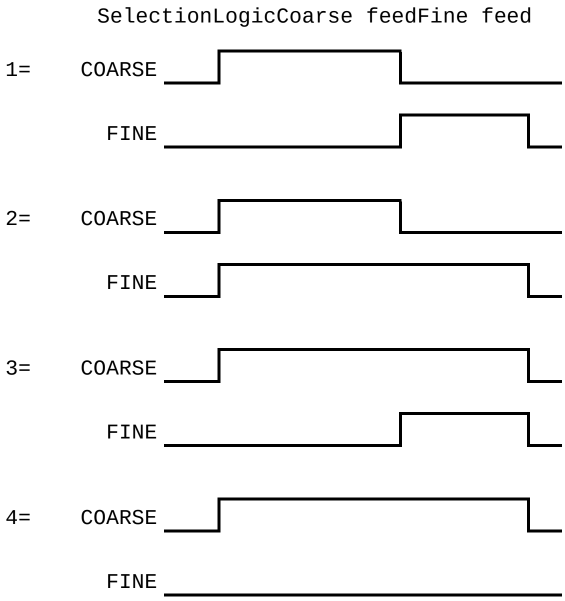
Call SCALEDOS (C1, C2)

Parameter	Significance	Permissible values
C1	Selection of multiple COARSE/FINE	0 - 1
C2	Number of pairs of COARSE/FINE. If parameter C1 is set to 0, parameter C2 must also be set to 0. If parameter C1 is set to 1, parameter C2 must have a value between 2 and 20.	0.2 - 20

Terminal Description

No	Name	Type	Description
F1	ADDR	II	**ADDR**ess. Strapped I/O address on DSXW 110/111 in decimal code. The board must be strapped within the interval for Misc Board which is address 192 to 239.
1	START	IB	**START**. Activates batching.
2	HOLD	IB	**HOLD**. Freezes current batching.
3	ABORT	IB	**ABORT**. Aborts current batching.
4	TARENET	IB	**TARE NET**. Tares net weight.
5	DEMWT	IR	**DEM**anded **W**eigh**T**. Set point for batching.

Terminal Description

No	Name	Type	Description
6	STOPLIM	IR	**STOP LIM**it. Stop limit for batching.
7	FINELIM	IR	**FINE LIM**it. Limit for transition of coarse/fine feed for batching.
8	HYST	IR	**HYST**eresis. Hysteresis value for the STOPLIM and FINELIM limits. The hysteresis is one sided, that is, when a limit has been reached, the value must fall below the limit by the hysteresis value for the corresponding signal (COARSE, FINE) to be activated again.
9	DOSSEL	IB	**DOS**ing **SEL**ect. Choice of type of batching. **Type** **Dosing** 0 Dosing in (filling) 1 Dosing out (emptying)
10	WTSEL	IB	**W**eigh**T** **SEL**ect. Selection of weight for batching. **Selection** **Dosing** 0 Gross weight 1 Gross weight
11	LOGSEL	II	**LOG**ic **SEL**ect. Selection of logic on batching.  SelectionLogicCoarse feedFine feed 1= COARSE FINE 2= COARSE FINE 3= COARSE FINE 4= COARSE FINE
12	DOACTV	IB	**D**igital **O**utputs **AC**Ti**V**e. Defines whether the digital outputs (2) are to be controlled by COARSE and FINE. **Selection** **Function** 0 Digital outputs are not used 1 Digital outputs give COARSE/FINE

Terminal Description

No	Name	Type	Description
13	ERR	OB	ERRor. Indicates error.
14	ERRTYPE	OI	ERRor TYPE. Indicates type of error.
15	DOSING	OB	**DOSING**. Indicates that batching is in progress.
16	STABLE	OB	**STABLE**. Indicates stable scale, that is, the weight change between two measured values is below the limit for stable rate (ΔWT<WTSTABL). **NOTE:** WTSTABL is defined by the SCALE element.
17	COARSE	OB	**COARSE**. Indicates that coarse feed is in progress. **NOTE:** Logic controlled by the choice of LOGSEL.
18	FINE	OB	**FINE**. Indicates that fine feed is in progress. **NOTE:** Logic controlled by the choice of LOGSEL.
20	BINSEL	II	**BIN SEL**ect. Selection of bin/material for batching.
30	COARSE1	OB	**COARSE1**. Indicates coarse feed for bin/material 1 selected via BINSEL. **NOTE:** Logic controlled by the choice of LOGSEL.
31	FINE1	OB	**FINE1**. Indicates coarse feed for bin/material 1 selected via BINSEL. **NOTE:** Logic controlled by the choice of LOGSEL.
32	COARSE2	OB	**COARSE2**. Indicates coarse feed for bin/material 2 selected via BINSEL. **NOTE:** Logic controlled by the choice of LOGSEL.
33	FINE2	OB	**FINE2**. Indicates coarse feed for bin/ material 2 selected via BINSEL. NOTE: Logic controlled by the choice of LOGSEL.
30+*C2*×2-2	COARSEC2	OB	**COARSEC2**. Indicates coarse feed for bin/material C2 selected via BINSEL. **NOTE:** Logic controlled by the choice of LOGSEL.
30+*C2*×2-1	FINEC2	OB	**FINEC2**. Indicates coarse feed for bin/material C2 selected via BINSEL. **NOTE:** Logic controlled by the choice of LOGSEL.

Function, General

The SCALEDOS element is used to connect batching data/signals produced on a weighing board, for example, DSXW 110/ 111, to PC programs. One SCALEDOS element is needed for every batching scale to be included. The SCALEDOS element requires the SCALE element to be defined and of course the presence of a weighing board, for example, DSXW 110/ 111.

A batching procedure goes like this: the batching scale is tared, batching starts, and the difference between actual weight (AW) and desired batching weight (set point BV) is compared with two limits (fine feed limit (FL) and stop limit (SL). The result of these comparisons influences two batching signals, coarse feed and fine feed. These signals control the feeders of the batching scale.

When DW-AW>FL, coarse feeding operates. When SL<DW-AW<FL, fine feeding operates. When DW-AW<SL, feeding stops. The stop limit (<>0) actually means a stop in advance, and this compensates for material on route, that is, material that has left a supply container but not yet reached the scale and influenced AW. The limits have one sided hysteresis to prevent the batching signals oscillating at the limit transitions, that is, when DW-AW falls below SL, DW-AW must exceed SG by the hysteresis value for the corresponding batching signal to be activated again.

Call parameter C1 allows a multiplexer function to be added for multiple COARSE/FINE batching signals. This is used when batching is to take place from more than one container. Call parameter C2 specifies the number of material containers and thus the number of pairs of COARSE/FINE.

Error codes

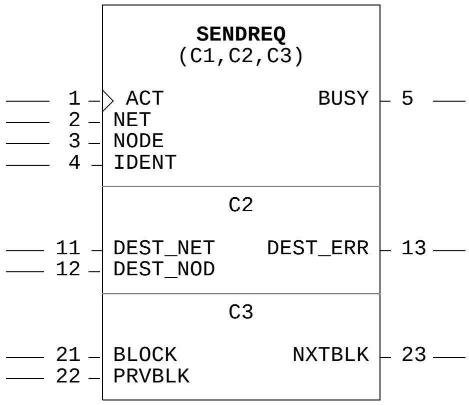
The outputs ERR and ERRTYPE specify the error states that can occur. ERR is a logic output that is set to 1 when an error is detected in the transmission of data from/to the weighing board. The output ERRTYPE mirrors the error state with a code as shown in the table below. The error codes are the same as indicated by the SCALE element.

Error	Error Description Code
–2	Calibration data absent, scale not calibrated
–1	Communication with weighing board not initiated
0	No error
1	Incorrect ’BOARD TYPE’ specified in data base
2	Bus malfunction
3	Fault in communication with weighing board
4	Error in initiation of weighing board
5	Weighing board missing
6	Data fault, SCALEDOS (BINSEL)
7	Data fault, SCALEDOS (LOADSEL)
11	Hardware fault, ±15 V stabilizer
12	Hardware fault, load cell supply
13	Hardware fault, A/D converter measurement references
14	Hardware fault, A/D converter test reference
15	Hardware fault, A/D converter counter
16	Hardware fault, ”time out” for A/D conversion
17	Measured value error, overflow in measurement channel offset
18	Measured value error, overflow in test reference
19	Measured value error, overflow in reference quotient
20	Measured value error, overflow in weight value
21	Address fault on PC element

Request Execution of Data Set **SENDREQ**

Summary

SENDREQ is used to request event controlled execution of Data Base Elements of type: Data Set Descr (DS), MVI Data Set Descr (MS), TEXT Data Set Descr (TS) or MVI Data Block (MVB)



PC element SENDREQ

Call SENDREQ (C1, C2, C3)

Parameter	Significance	Permissible values
C1	Type of Data Set (1=DS, 2=MS, 3=TS, 4(1)=MVB)	1 -.4(1)
C2	Connection to other node. 0=Data Set element is in the same node. 1=Data Set element is in another node, connections 11, 12 and 13 are used.	0 -.1
C3	Connection to block sending. 0=Sending cannot be blocked. 1=Sending can be blocked, connections 21, 22 and 23 can be used.	0 - 1

(1) C1=4 is only valid for Advant Controller 400 Series **1.2 and later**.

Terminal Description

No	Name	Type	Description
1	ACT	IB	**ACT**ivate. The Data Set element with the identity given by NET, NODE and IDENT is executed once when ACT goes from 0 to 1.
2	NET	II	**NET**. The network specified in the NET parameter of the Data Set element.
3	NODE	II	**NODE**. The node number specified in the node parameter of the Data Set element
4	IDENT	II	**IDENT**ity. The identity specified in the IDENT parameter of the Data Set element.
5	BUSY	OB	**BUSY**. The element is busy with sending.
11	DEST_NET	II	**DEST**ination**_NET**. Network for the Data Set element when this is not in the same node as the SENDREQ element.
12	DEST_NOD	II	**DEST**ination**_NODE**. Node for the Data Set element when this is not in the same node as the SENDREQ element.

Terminal Description

No	Name	Type	Description
13	DEST_ERR	OB	**DEST**ination**_ERR**or. The output is set true of the sending node does not establish contact with the node specified by DEST_NODE and DEST_NET.
21	BLOCK	IB	**BLOCK**. No sending takes place while BLOCK is true. If sending orders are given at the ACT terminal during blocking by BLOCK, they are ignored.
22	PRVBLK	IB	**PR**e**V**ious **BL**oc**K**. Used together with NXTBLK to control the flow to the Data Set elements. No sending begins while PRVBLK is set to true. If sending orders are given at the ACT terminal during blocking by PRVBLK, the data package corresponding to last given order will be sent when PRVBLK is set to false.
23	NXTBLK	OB	**N**e**XTBL**oc**K**. NXTBLK is true when sending is in progress, until sending is complete, when it is set to false. When no sending is in progress, NXTBLK follows the value of PRVBLK. Flow control to the Data Set elements is executed by connecting NXTBLK to the PRVBLK input of the next SENDREQ element. The NXTBLK input of this element is then connected to the PRVBLK input of the third SENDREQ element and so on. The NXTBLK output of the last element is connected to the first element, thus the SENDREQ elements are connected in a chain.

Function

The element is executed when the ACT input goes from 0 to 1. The address of the Data Set element (DS, MS, TS, MVB[1]) to be activated is given by NET, NODE and IDENT. If the Data Set element is in a different node from the SENDREQ element, the address is given by DEST_NODE and DEST_NET.

Sending takes place regardless of the value of associated Data Base element terminal BLOCKED (only DS, MS, and MVB[1] have this parameter). Sending can be blocked with the BLOCK or PRVBLK terminal of the SENDREQ element. If sending orders, received by ACT during blocked state, shall be effected afterwards the terminal PRVBLK should be used for blocking else terminal BLOCK should be used for blocking.
By linking several SENDREQ elements to each other with NXTBLK and PRVBLK, the order in which the Data Set elements are to be activated can be controlled.

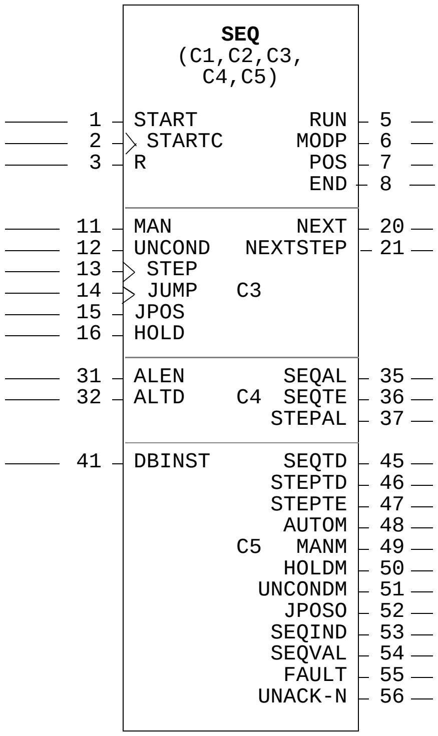
---

[1] MVB is only valid for Advant Controller 400 Series **1.2 and later**.

Sequence Header SEQ

Summary

SEQ is used to control a sequence.



PC element SEQ

Call SEQ (C1, C2, C3, C4, C5)

Parameter	Significance	Permissible values
C1	Cyclicity in ms	Permissible values are determined by the possible cycle times of the relevant PC system. These times are specified in the manual for the system concerned.
C2	Place in the cycle time table	Permissible values are determined by the relation to other execution units with the same cycle time. The maximum value is determined by the number of possible places with the same cycle time. The system selects a free place with the lowest number if no value is specified at entry.
C3	Selection of manual control	0 - 1
C4	Selection of time function	0 - 1
C5	Selection of MMC function	0 - 1

Terminal Description

No	Name	Type	Description
1	START	IB	Input for start of sequence.
2	STARTC	IB	**START C**lock. Dynamic input for start of the sequence. Used when the sequence is to be run through once only.
3	R	IB	**R**eset. Clearing of the elements of the sequence.
5	RUN	OB	Indicates that the sequence is in operation.
6	MODP	OB	**MOD**ify **P**ermission. Specifies if the sequence may be modified from the engineering station.
7	POS	OI	**POS**ition. Output which specifies the step number of the active step.
8	END	OB	Output which is active when the last step of the sequence is active.
11	MAN	IB	**MAN**ual. Input which is to be set for manual stepping of the sequence.
12	UNCOND	IB	**UNCOND**itional. Input which permits stepping even when the condition for the next step is not satisfied.
13	STEP	IB	Input for step change.
14	JUMP	IB	Input for jump to step specified by JPOS.
15	JPOS	II	**J**ump **POS**ition. Input which specifies the last part of the record designation for the step to which the jump is to be made.
16	HOLD	IB	Input which is used to stop the execution.
20	NEXT	OB	Output which is set when any condition for step change or jump is satisfied or when UNCOND is set.
21	NEXTSTEP	OI	Step to which stepping in manual mode is performed. Is connected to the corresponding input of the data base element when MMC function is used.
31	ALEN	IB	**AL**arm **EN**able. Input for start and stop of time measurement.
32	ALTD	IT	**AL**arm **T**ime **D**elay. Input for time after which SEQAL is to be set.
35	SEQAL	OB	**SEQ**uence **AL**arm. Output which is set when the time specified on ALTD has elapsed.
36	SEQTE	OT	**SEQ**uence **T**ime **E**lapsed. Output for elapsed time.
37	STEPAL	OB	**STEP AL**arm. Output for summary signal which indicates that the time set in an active subordinate step has elapsed.
41	DBINST	II	**D**ata **B**ase **INST**ance. The instance number of the sequence in the data base index, alternatively the name of the instance.
45	SEQTD	OT	**SEQ**uence **T**ime **D**elay. Output that indicates ALTD.
46	STEPTD	OT	**STEP T**ime **D**elay. Output giving ALTD for the active step.
47	STEPTE	OT	**STEP T**ime **E**lapsed. Output for time that has elapsed in the timer of the active step.
48	AUTOM	OB	**AUTO M**ode. Output which is set when operating in mode AUTO.
49	MANM	OB	**MAN**ual **M**ode. Output which is set when operating in mode MAN.

Terminal Description

No	Name	Type	Description
50	HOLDM	OB	**HOLD M**ode. Output which is set when operating in mode MAN.
51	UNCONDM	OB	**UNCOND**itional **M**ode. Output which is set when operating in mode MAN with UNCOND.
52	JPOSO	OI	**J**ump **POS**ition **O**utput. Output that indicates the jump target JPOS.
53	SEQIND	OI	**SEQ**uence **IND**ications. Output for status information to the data base element.
54	SEQVAL	OA20	**SEQ**uence **VAL**ues. Output for value information to the data base element.
55	FAULT	OI	Output for error information to the data base element.
56	UNACK-N	OB	**UNACK**nowledged**-N**. Output to the data base element for clearing of dialogue. The dialogue is cleared with each execution.

Function

The sequence header SEQ is a superior execution controlling element for a number of sequence steps. How often the sequence is to be executed is specified with the call parameter C1 and the place in the cycle time table on which the order of execution of the execution units with the same cycle time is based is specified with the call parameter C2. The basic function can be extended with the call parameters C3, C4 and C5. Manual control of the sequence can be obtained by setting C3 to 1 at call. A time function can be obtained by setting C4 to 1. The functions necessary for the element to be able to work together with MMC (Man Machine Communication) functions can be obtained by setting the call parameter C5 to 1. For C5 to be selected, the functions according to the call parameters C3 and C4 must be selected. In the basic version for automatic stepping only, and with no time or MMC functions, parameters C3, C4 and C5 must be 0.

Start of Sequence

With normal execution, the sequence can be started with a pulse at the input START. The pulse is registered in an internal memory in the sequence header. If a fixed 1 is connected to START, the sequence can be connected for automatic restart after resetting. If the sequence is to be run through only once, it must be started with STARTC. STARTC is a dynamic input and thus the internal memory of the sequence header is only affected when the input status is changed from 0 to 1. After a start, the output RUN is set and the condition input for the first step is checked. If the condition is satisfied, the first step and its elements are executed.

Clearing

If the input R is set, the sequence is executed in the reset mode. This means that all outputs on the elements within all of the steps of the sequence are given initial values, which in most cases is the 0-value of the data. Reading to the data base is done with the data value 0 for all data types. For the steps, reset execution is performed only during the cycle when R goes to 1. For the sequence header, reset execution is performed during each cycle while R is set to 1. Input R overrides the inputs START, STARTC and HOLD.

RUN

The output RUN is set when the sequence is under execution, that is, when started with START or STARTC. RUN is cleared by R, HOLD and blocking functions introduced from service aids.

MODP

The sequence header contains a built in check of modification rights, MODP. If the modification rights are deactivated (MODP=0), the possibility of introducing changes in the function of the sequence from service aids is blocked. However, blocking can still be entered and operational parameters can be changed. The modification rights are activated and deactivated from service aids with special commands.

POS and END

The output POS specifies, with an integer (I), the last part of the record designation of the sequence step which is active. If POS is 0, the sequence header is active. The output END is set when the last step in the sequence is active.

Manual Execution

For manual stepping of a sequence, the input MAN must be set to 1. The sequence is started in the same way as with automatic stepping by setting START or STARTC to 1. When the condition for step change, or a jump condition, is satisfied, the output NEXT is set, and the step change can be performed with a pulse at the input STEP. Jumps have priority to ordinary step change.If stepping is required without the condition for the next step being satisfied, the input UNCOND must be set to 1 when STEP is set to 1.

Jump in Sequence

A jump can be obtained by setting the input JUMP. The jump is made to the step which is specified at the input JPOS. The step is specified by the last part of its record designation. The jump is performed irrespective of the status at the inputs MAN, STEP and UNCOND. If the MMC function is used and a jump is to be initiated by the operator via the dialogue, the operation mode must be MANM. A jump is not possible if HOLD or R is set.

HOLD

When HOLD is set, execution of active sequence steps is stopped. The calculated data remain until the next time the sequence step is executed. The execution is stopped and the output RUN is reset while HOLD is set to 1. The RUN output at the step header remains set. HOLD also stops the timers in the sequence header and the step headers.

Time Functions

If the call parameter C4 is given as 1, a time function associated with the sequence header is obtained. A common signal STEPAL is obtained which indicates any time function which has elapsed in any of the steps. The switch on of the timer in the sequence header is delayed with accumulating time measurement controlled by the input ALEN. If ALEN is reset, the timer is stopped but its contents remain and when ALEN is set, the time measurement restarts from that value which applied when ALEN was reset. The setting of HOLD has the same effect on the timer as resetting ALEN. The timer can only be started when the sequence is active.

When the time specified at the input ALTD has elapsed, the output SEQAL is set. When ALTD is set equal to 0, SEQAL is set to 1 irrespective of the state at input ALEN. The output SEQTE gives the time which has elapsed with the timer active. The timer is cleared when the complete sequence is cleared with the input R. A timer corresponding to that in the sequence header can be selected for the sequence steps. The output STEPAL, which specifies if the time set in the timer for the active step has elapsed, is located in the sequence header. STEPAL is set to 1 if the delay specified for the step concerned has elapsed.

MMC (Man Machine Communication) Functions

If the call parameter C5 is given as 1, connections mainly used for an MMC function are obtained. See the *Functional Units manuals* for further information.

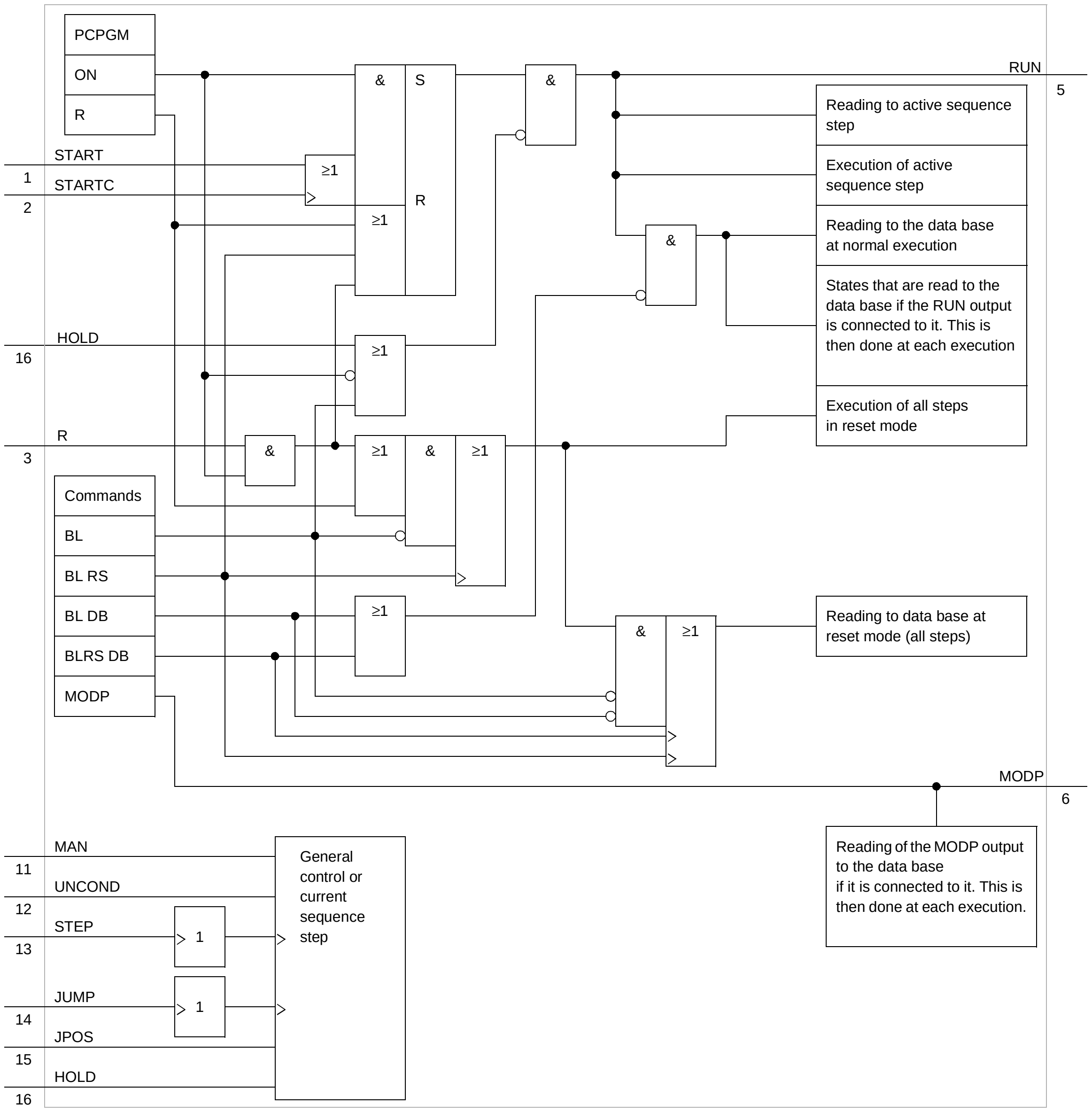
Blocking

The execution of the steps controlled by the sequence header can be blocked in several different ways from a service aid.

- The command BL <module identity>(**BL**ock) blocks all execution of the sequence including data exchange with the data base.
- The command BL DB,<module identity>(**BL**ock **D**ata **B**ase) blocks reading to the data base of the data base variables occurring in the sequence.
- The command BLRS <module identity>(**BL**ock and **R**e**S**et) first clears the sequence and data to the data base (Reset execution) and then blocks all execution.
- The command BLRS DB,<module identity>(**BL**ock and **R**e**S**et **D**ata **B**ase) first clears the output data to the data base for the sequence and then blocks all further reading.
- The command DBL<module identity>(**D**e**BL**ock) is used to cancel blockings.

Effects from PCPGM

The ON and R inputs and the blocking functions in the program header override the inputs on the control module header. The input ON on PCPGM must be set to permit normal execution or reset execution of the step elements in the sequence.

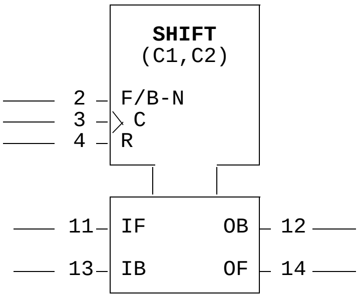

PCPGM
ON
R
START
1
STARTC
2
HOLD
16
R
3
Commands
BL
BL RS
BL DB
BLRS DB
MODP
&
S
R
≥1
RUN
5
Reading to active sequence step
Execution of active sequence step
Reading to the data base at normal execution
States that are read to the data base if the RUN output is connected to it. This is then done at each execution
Execution of all steps in reset mode
Reading to data base at reset mode (all steps)
MODP
6
Reading of the MODP output to the data base if it is connected to it. This is then done at each execution.
MAN
11
UNCOND
12
STEP
13
JUMP
14
JPOS
15
HOLD
16
1
General control or current sequence step

Function diagram

Shift Register SHIFT

Summary

Shift register SHIFT is used as a shift register of optional length. Data can be integers, real numbers, Boolean or time.



PC element SHIFT

Call SHIFT (C1, C2)

Parameter	Significance	Permissible values
C1	Data type	I, IL, R, B, T, TR
C2	Number of places	2 - 250

Terminal Description

No	Name	Type	Description
2	F/B-N	IB	**F**orward/**B**ackward**-N**. Input for selection of shift direction. Shifting is forward when F/B-N=1 and backward when F/B-N=0.
3	C	IB	**C**ontrol. Dynamic input for shifting.
4	R	IB	**R**eset. Input for clearing the shift register. All further shifting is prevented when R=1.
11	IF	I*C1*	**I**nput **F**orward. Input data with forward shift.
12	OB	O*C1*	**O**utput **B**ackward. Output data with backward shift.
13	IB	I*C1*	**I**nput **B**ackward. Input data with backward shift.
14	OF	O*C1*	Output Forward. Output data with forward shift.

Function, Forward Shift

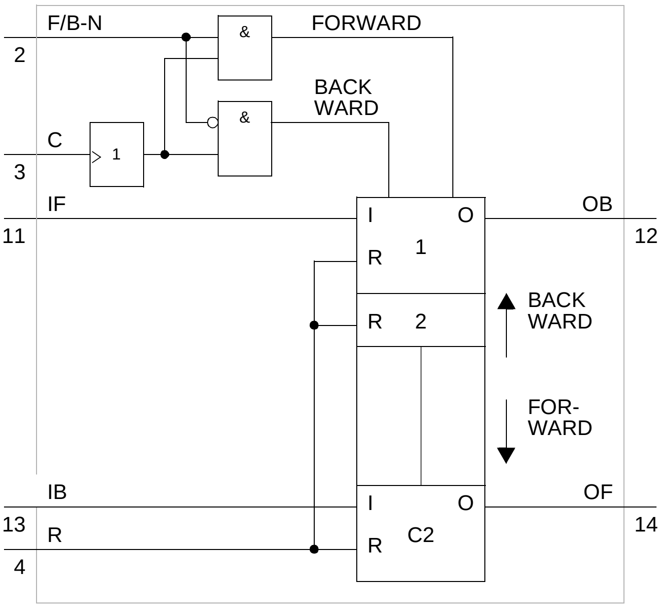
When input C goes to 1 and the input F/B-N is 1, data is shifted forwards in the register and data is read into position 1 from input IF. Data which before the shifting was located at output OF, is replaced at the shift forward by data at the position next to last.

Backward Shift

When input C goes to 1 and the input F/B-N is 0, data is shifted backwards in the register and data is read into position C2 from input IB. Data which before the shifting was located at output OB, is replaced at the shift backward by data from position 2.

Clearing

The input R clears the shift register and prevents all further reading into the shift register.

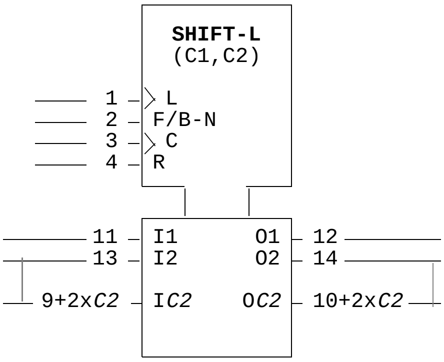


Function diagram

Shift Register with Loading SHIFT-L

Summary

SHIFT-L (**SHIFT - L**oad) is used as a shift register of optional length. All positions can be loaded and read. Data can be integers, real number, Boolean or time.



PC element SHIFT-L

Call SHIFT-L (C1, C2)

Parameter	Significance	Permissible values
C1	Data type	I, IL, R, B, T, TR
C2	Number of places	1 - 35

Terminal Description

No	Name	Type	Description
1	L	IB	**L**oad. Dynamic input for loading the shift register.
2	F/B-N	IB	**F**orward/**B**ackward**-N**. Input for selection of shift direction. Shifting is forward when F/B-N=1 and backward when F/B-N=0.
3	C	IB	**C**ontrol. Dynamic input for shifting.
4	R	IB	**R**eset. Input for clearing the shift register. All further entry and shifting is prevented when R=1.
11	I1	I*C1*	**I**nput **1**. Input data to position 1.
12	O1	O*C1*	**O**utput **1**. Output data from position 1.
13	I2	I*C1*	**I**nput **2**. Input data to position 2.
14	O2	O*C1*	**O**utput **2**. Output data from position 2.
: 9+2x*C2*	I*C2*	I*C1*	**I**nput ***C2***. Input data to position C2.
10+2x*C2*	O*C2*	O*C1*	**O**utput ***C2***. Output data from position C2.

Function, Loading

When the input L is set (to 1), data is read in from inputs I1 - I*C2*, after which the data entered can be read at outputs O1 - O*C2*. If input C and input L are set at the same time, data is first read into the register. Shifting is then performed as follows.

Forward Shift

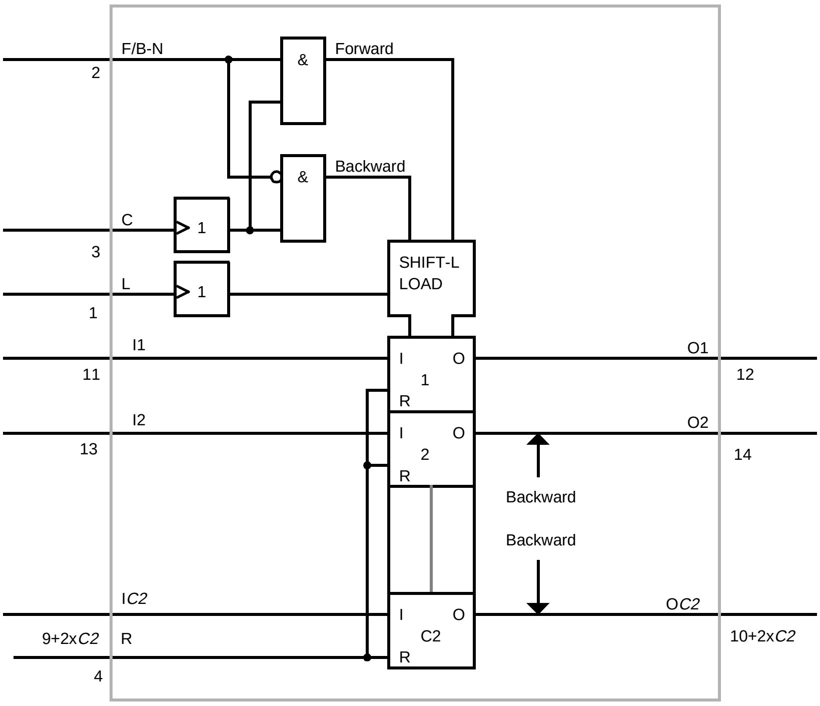
When input C goes to 1 and the input F/B-N is 1, data is shifted forwards in the register and data is read into position 1 from input I1.

Backward Shift

When input C goes to 1 and the input F/B-N is 0, data is shifted backwards in the register and data is read into position C2 from input I*C2*.

Clearing

The input R clears the shift register and prevents all further loading.



Function diagram

Slave Header SLAVEM

Summary

SLAVEM (**SLAVE M**odule) is used for execution control of a slave module.

```
SLAVEM (C1, C2)
```

PC element SLAVEM

Call SLAVEM (C1, C2)

Parameter	Significance	Permissible values
C1	Identity of execution controlling master header	The second part of the PC record designation of the master header.
C2	Currently not used	

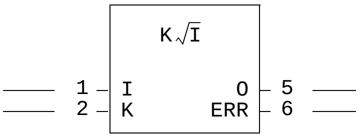
Function

The slave header SLAVEM is an element header for a slave module. The numerical identity of the master header associated with the slave module is specified with the call parameter C1. This then becomes the second part of the PC record designation of the master as this must always be at the level directly subordinate to a program header. How often and in what order the slave module is to be executed in relation to other execution units is determined by the superior master header. The slave modules within the same master are executed in the order of the documentation. Blocking from service aids and PC programs is determined entirely by the master.

Square Root SQRT

Summary

SQRT (**SQ**uare **R**oo**T**) is used to calculate the square root of a real number. The result can be multiplied by an optional value.



PC element SQRT

Call SQRT

Terminal Description

No	Name	Type	Description
1	I	IR	**I**nput. Input for number of which the square root is to be calculated.
2	K	IR	Input for multiplication factor K.
5	O	OR	**O**utput. Output for the product of the value at input K and the square root of the value at input I.
6	ERR	OB	**ERR**or. Output which is set if the value at input is negative.

Function

The square root of the value at input I is calculated. The result is multiplied by the value at input K. The product is stored at the output O.

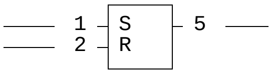
Supervision

If the value at the input I is negative, the error output ERR is set to 1. The value 0.0 is then stored at the output O.

Set/ Reset Memory SR

Summary

SR (**S**et **R**eset memory) is used as a memory for Boolean variables.



PC element SR

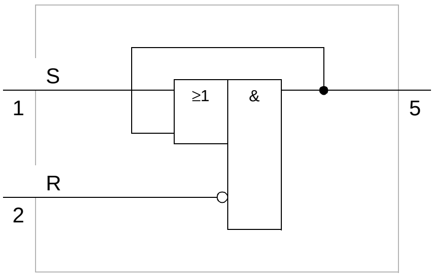
Call SR

Terminal Description

No	Name	Type	Description
1	S	IB	**S**et input.
2	R	IB	**R**eset input which overrides the set input.
5	–	OB	Output from the memory element.

Function

The output is set (to 1) if the set input is set at the same time that the reset input is reset (to 0). If the reset input is set, the output is unconditionally reset.

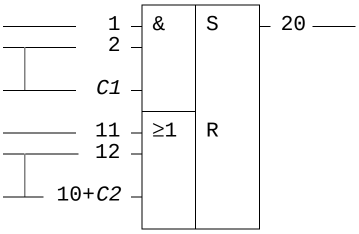


Function diagram

Memory Elements SR-AA, SR-AO, SR-OO, SR-OA

Summary

SR-AA, SR-AO, SR-OA and SR-OO (**S**et **R**eset memory - **A**=And, **O**=Or) are used as a memory for Boolean variables. The elements have different gate elements for condition generation at the set and reset inputs.



PC element SR-AO

Call SR-AA (C1, C2), SR-AO (C1, C2), SR-OA (C1, C2) SR-OO (C1, C2)

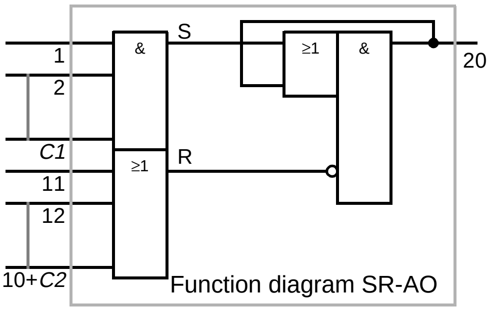
Parameter	Significance	Permissible values
C1	Number of inputs in the set gate	1 - 9
C2	Number of inputs in the reset gate	1 - 9

Terminal Description

No	Name	Type	Description
1	–	IB	Input to the set function.
2	-	IB	Input to the set function.
: *C1*	-	IB	Input to the set function.
11	-	IB	Input to the reset function.
12	-	IB	Input to the reset function.
: 10+*C2*	-	IB	Input to the reset function.
20	-	OB	Memory element output.

Function

The output is set to 1 if the conditions for the set gate (AND or OR) are satisfied at the same time that the conditions for the reset gate are not satisfied. If the conditions for the reset gate are satisfied, the output is unconditionally reset.

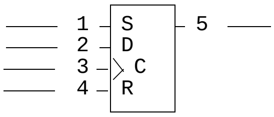


Function diagram SR-AO

Set/ Reset Memory with Data Input SR-D

Summary

SR-D (**S**et **R**eset memory - **D**ata input) is used as a memory for Boolean variables. Besides the SR-function, it can also clock in data.



PC element SR-D

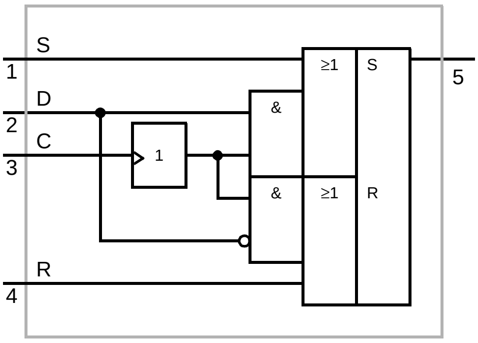
Call SR-D

Terminal Description

No	Name	Type	Description
1	S	IB	**S**et input.
2	D	IB	**D**ata input.
3	C	IB	**C**lock. Dynamic input for entry of data from the D input.
4	R	IB	**R**eset input which overrides all other inputs.
5	-	OB	Output from the memory element.

Function

If only the S and R inputs are used, SR-D functions as an ordinary SR element. When input R is reset and input C goes to 1, the value at input D is stored at output 5. When input R is set, output 5 is unconditionally reset, that is, R overrides the other inputs.

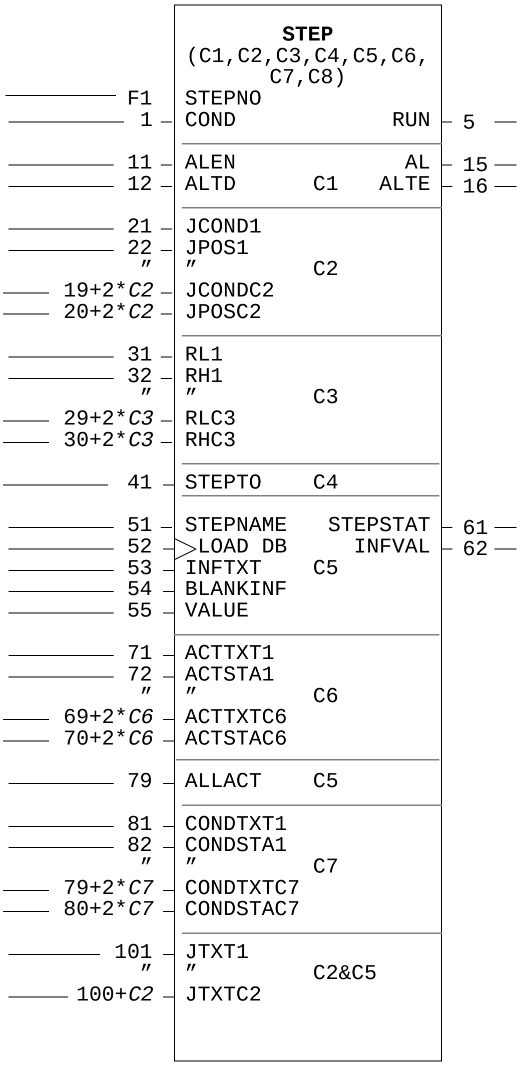


Function diagram

Step Header STEP

Summary

STEP is used to partition a sequence into sequence steps.



PC element STEP

Call STEP (C1, C2, C3, C4, C5, C6, C7, C8)

Parameter	Significance	Permissible values
C1	Selection of time function	0 - 1
C2	Number of jumps	0 - 4
C3	Number of reset groups	0 - 4
C4	Selection of alternative execution order	0 - 1

Parameter	Significance	Permissible values
C5	Selection of MMC function	0 - 1
C6	Number of activities	0 - 4
C7	Number of step conditions	0 - 8
C8	Maximum number of characters in the texts.	0 - 15
If C5=0 then must C6=C7=C8=0 If C5=1 then must C8>0		

Terminal Description

No	Name	Type	Description
F1	STEPNO	FI	**STEP N**umber. Defines the step number of the step element. If STEPNO is anything but zero, STEPNO is used as the step number. If STEPNO is zero, the last part of the item designation of the STEP element is used as the step number. The STEP number is not used for normal sequence execution, only as input to alternative execution order (for example, as input to the JPOS and STEPTO terminals). Range: 0 - 999.
1	COND	IB	**COND**ition. Condition input for stepping.
5	RUN	OB	Indicates that the step is active.
11	ALEN	IB	**AL**arm **EN**able. Input for start and stop of time measurement.
12	ALTD	IT	**AL**arm **T**ime **D**elay. Input for alarm time.
15	AL	OB	**AL**arm. Output for alarm. Set (to 1) when the time specified at the input ALTD has elapsed.
16	ALTE	OT	**AL**arm **T**ime **E**lapsed. Output for elapsed time.
21	JCOND1	IB	**J**ump **COND**ition **1**. Input for jump to step specified by JPOS1.
22	JPOS1	II	**J**ump **POS**ition **1**. Specifies to which step the jump is to be made if JCOND1 is set.
23	JCOND2	IB	**J**ump **COND**ition **2**. Input for jump to step specified by JPOS2.
24	JPOS2	II	**J**ump **POS**ition **2**. Specifies to which step the jump is to be made if JCOND2 is set.
: 19+2x*C2*	JCOND*C2*	IB	**J**ump **COND**ition ***C2***. Input for jump to step specified by JPOS*C2*.
20+2x*C2*	JPOS*C2*	II	**J**ump **POS**ition ***C2***. Specifies to which step the jump is to be made if JCOND*C2* is set.
31	RL1	II	**R**eset **L**ow limit **1**. Input for the number of the first step in the first group of steps to be cleared.
32	RH1	II	**R**eset **H**igh limit **1**. Input for the number of the last step in the first group of steps to be cleared.
33	RL2	II	**R**eset **L**ow limit **2**. Input for the number of the first step in the second group of steps to be cleared.
34	RH2	II	**R**eset **H**igh limit **2**. Input for the number of the last step in the second group of steps to be cleared.

Terminal Description

No	Name	Type	Description
: 29+2x*C3*	RL*C3*	II	**R**eset **L**ow limit ***C3***. Input for the number of the first step in the last group of steps to be cleared.
30+2x*C3*	RH*C3*	II	**R**eset **H**igh limit ***C3***. Input for the number of the last step in the last group of steps to be cleared.
41	STEPTO	II	**STEP T**O. This input modifies the execution order of the sequence. The input determines which step should be next step in the sequence. To be used in convergence of a sequence selection.
51	STEPNAME	IA12	**STEP NAME**. Name of the sequence step. Max 12 characters.
52	LOAD_DB	IB	**LOAD D**ata **B**ase. Dynamic input that updates the data base with all texts in the element. This input should be used only if you want to change some of the texts during execution of the step.
53	INFTXT	IA*C8*	**INFO**rmation **T**e**XT**. Information text for the sequence step.
54	BLANKINF	IB	**BLANK INF**ormation text. Input set if no information text is to be displayed.
55	VALUE	IR	Information **VALUE**. Input for value to Operator Station.
61	STEPSTAT	OIL	**STEP STATUS** output. Connection to the data base element.
62	INFVAL	OR	Value output. Connection to the data base element.
71	ACTTXT1	IA*C8*	**ACT**ivity **T**e**XT 1**. Text describing activity 1.
72	ACTSTA1	IB	ACTivity STAtus 1. Input determining status of activity 1.
69+2x*C6*	ACTTXT*C6*	IA*C8*	**ACT**ivity **T**e**XT C6**. Text describing activity C6 in the sequence step.
70+2x*C6*	ACTSTA*C6*	IB	**ACT**ivity **STA**tus **C6**. Input determining status of activity C6.
79	ALLACT	IB	**ALL ACT**ivities. Input confirming that all activities in the sequence step has been executed.
81	CONDTXT1	IA*C8*	**COND**ition **T**e**XT 1**. Text describing step condition 1.
82	CONDSTA1	IB	**COND**ition **STA**tus **1**. Input determining status of step condition 1.
: 79+2x*C7*	CONDTXT*C7*	IA*C8*	**COND**ition **T**e**XT C7**. Text describing step condition *C7.*
80+2xC7	CONDSTA*C7*	IB	**COND**ition **STA**tus **C7**. Input determining status of step condition *C7.*
101	JTXT1	IA*C8*	**J**ump **T**e**XT 1**. Text describing jump condition 1.
: 100+*C2*	JTXT*C2*	IA*C8*	**J**ump **T**e**XT C2**. Text describing jump condition *C2.*

Function

STEP is a superior element header for partitioning a sequence into sequence steps. Elements, function modules and blocks can be directly subordinate to a step header. How often and in which order the step is to be executed in relation to other execution units is determined in the superior sequence header. The element has a basic function which may be extended by setting one or more of the call parameters C1, C4 to C8 to a value different from 0. A time function can be obtained with the call parameter C1. Jump and reset functions can be obtained with C2 and C3. With C4 an alternative execution order is obtained. C5-C8 means selection of extended MMC functions.

Basic Version

The basic version of STEP includes only the condition input COND for stepping and the output RUN which indicates that the step is active.

Normal Execution

If the sequence is executing normally in accordance with the conditions in the sequence header and the COND input of the following step is set to 1, a step change takes place. A step may also become active through a jump from the sequence header. A step may become active without COND being set to 1. This occurs when UNCONDitional on the sequence header is used.

With the step change, the RUN output of the active step is reset, after which the elements in this step are executed once in the normal way. The RUN output of the new active step is then set and the elements in this step are then executed. If the input HOLD on the superior sequence header is set to 1, execution of the active step is stopped and the RUN output remains set.

Time Function

If the call parameter C1 is specified as 1, a timer with accumulating time measurement, controlled by the input ALEN, is obtained. If ALEN is reset, the timer is stopped but the contents remain and when ALEN is set, the time measurement continues from the value which applied when ALEN was reset. Setting HOLD in the superior sequence header gives the same result on the time measurement circuit as the resetting of ALEN. The timer can only be started when the step is active. Elapsed time is specified at the output ALTE. When the time specified at the input ALTD has elapsed, the output AL is set. The timer is reset when the step is cleared by another step (see below) or by the setting of the R input in the superior sequence header.

Jump

When any of the condition inputs JCOND1 - JCOND*C2* are set, a jump is performed to the step specified at the corresponding input of JPOS1 - JPOS*C2*. If several condition inputs are set at the same time, a jump is made to the step specified by the input with the lowest sequential number. If the conditions for step change are satisfied at the same time as those for a jump, the jump is performed.

No jumps can be performed when the input HOLD on the superior sequence header is set. If the input MAN on the sequence header is set, a jump is not made until the input STEP is set. When executing the step, the elements included are executed first, after which the jump conditions are checked. If both the conditions for jump and those for step change are satisfied, the jump is made. If JPOS=0 for the jump input concerned, a jump is made to the superior sequence header. If the step specified by the JPOS input does not exist, the sequence stops.

Clearing of Step

When a sequence step has been active, the data calculated remains, provided that the RUN output is not included as a condition for those data. Calculated data can be cleared by a later sequence step. One or more steps in succession can be cleared simultaneously. The step from which clearing is to be performed is specified with an integer (I) on one of the inputs RL1 - RL*C3*. The step to which clearing is to be performed is specified with an integer (I) on the corresponding input of RH1 - RH*C3*. Clearing is performed when leaving the step with a normal step change or with a step change controlled by a jump condition. If RL=0, the complete sequence is cleared, which corresponds to a reset on the sequence header. In this case RH has no effect. If the RL value does not exist, there is no reset. If RH does not exist or has a value which is smaller than RL, the reset is performed to the end of the sequence.

Alternative Execution Order when Joining Branched Sequences

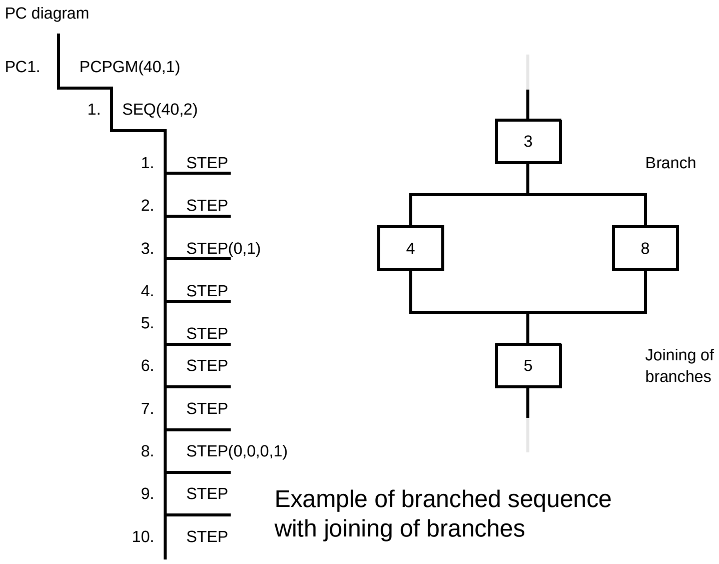
Normally, the steps are executed in the order they are build according to the PC diagram. When a sequence is branched, the jump function is used. However, when joining several sequence branches it is better to use the alternative execution order function.

Input STEPTO then determines which step should be next step in the sequence. This step will be next step in the sequence irrespective of where it is situated in the PC diagram. Several sequence branches can have the same step determined as next step. Thus it is possible to join several sequence branches to a common step irrespective of where the steps are situated in the PC diagram. If STEPTO=0 the execution is changed to a superior sequence head, and if STEPTO gives a non-existing step the sequence is interrupted.

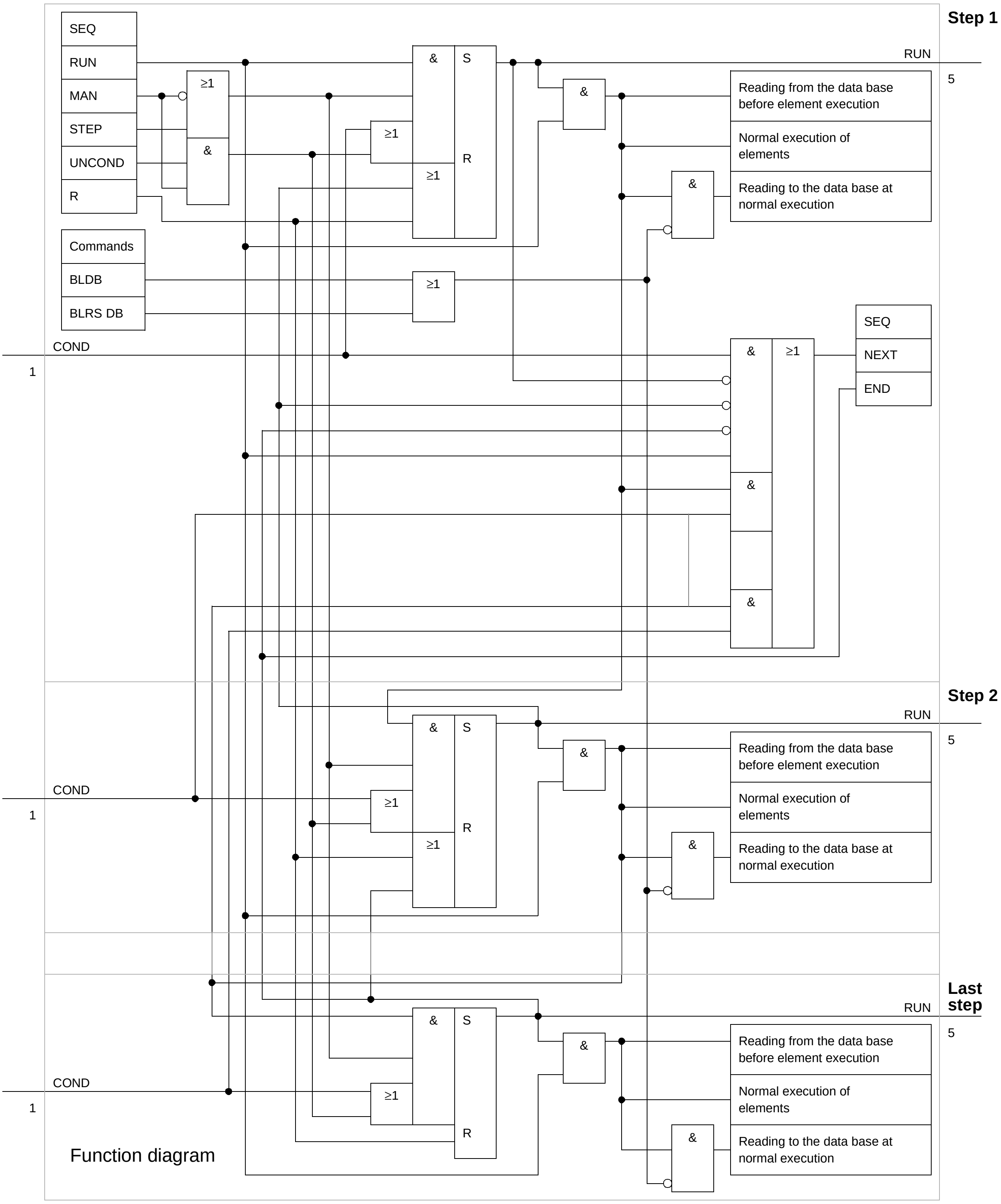
Example: In the branched sequence in the figure below, the left branch is build as a straight, normal sequence. Branching is performed with jumps (C2=1 and JPOS1=8 in step 3) and the joining of the branches is executed with alternative execution order (C4=1 and STEPTO=5 in step 8). Thus step 5 will follow step 8 when executing the sequence. The step following step 8 in the PC diagram (in this case step 9) will not affect the execution of step 8 in any way.

MMC Functions

Step head STEP is part of the functional unit SEQUENCE. Call parameters C5-C8 are then used to define different MMC functions. Further information can be obtained in the manual Functional Units part 3, SEQ, GROUP.



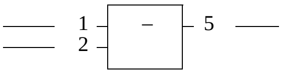
Example of branched sequence with joining of branches


Step 1
SEQ
RUN
MAN
STEP
UNCOND
R
≥1
&
&
S
R
≥1
≥1
&
RUN
5
Reading from the data base before element execution
Normal execution of elements
Reading to the data base at normal execution
&
Commands
BLDB
BLRS DB
≥1
SEQ
NEXT
END
&
≥1
&
&
COND
1
Step 2
RUN
5
&
S
≥1
≥1
R
&
&
Reading from the data base before element execution
Normal execution of elements
Reading to the data base at normal execution
COND
1
Last step
RUN
5
&
S
≥1
R
&
&
Reading from the data base before element execution
Normal execution of elements
Reading to the data base at normal execution
COND
1
Function diagram

Subtractor SUB

Summary

SUB is used for subtraction of two integers or real numbers.



PC element SUB

Call SUB (C1)

Parameter	Significance	Permissible values
C1	Data type	I, IL, R

Terminal Description

No	Name	Type	Description
1	–	I*C1*	Input for minuend.
2	–	I*C1*	Input for subtrahend.
20	–	O*C1*	Output for difference.

Function

The value at input 2 is subtracted from that at input 1 and the result is stored at output 20.

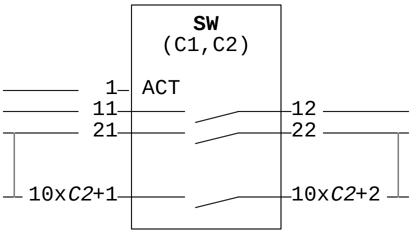
Overflow

If the maximum positive or negative values are exceeded, the output is limited to the highest or lowest representable value for the data type.

Switch SW

Summary

SW (**SW**itch) is used as a connection element for data and has an optional number of channels with closing function. The data type can be integer, real number, Boolean or time.



PC element SW

Call SW (C1, C2)

Parameter	Significance	Permissible values
C1	Data type	I, IL, R, B, T, TR
C2	Number of channels	1 - 9

Terminal Description

No	Name	Type	Description
1	ACT	IB	**ACT**ivate. Input for activation of the switch. When the input is set to 1 the switch is activated.
11	-	I*C1*	Input to channel 1 which is connected to the output for channel 1 when the switch is activated.
12	-	O*C1*	Output from channel 1.
21	-	I*C1*	Input to channel 2 which is connected to the output for channel 2 when the switch is activated.
22	-	O*C1*	Output from channel 2.
: 10x*C2*+1	-	I*C1*	Input to channel C2 which is connected to the output for channel C2 when the switch is activated.
10x*C2*+2	-	O*C1*	Output from channel C2.

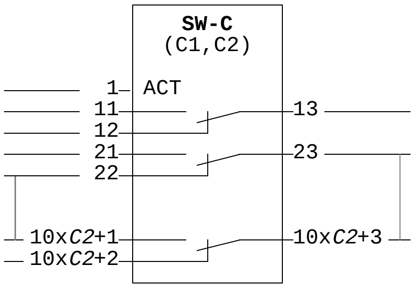
Function

When the control input ACT is 0, the output data is 0 according to the data type. When ACT is set, data comes from the inputs 11 - 10x*C2*+1.

Changeover Switch SW–C

Summary

SW-C (**SW**itch **- C**hangeover) is used as a connection element for data and has an optional number of channels with switching function. The data type can be integer, real number, Boolean or time.



PC element SW-C

Call SW-C (C1, C2)

Parameter	Significance	Permissible values
C1	Data type	I, IL, R, B, T, TR
C2	Number of channels	1 - 9

Terminal Description

No	Name	Type	Description
1	ACT	IB	**ACT**ivate. Input for activation of the switch. When the input is set to 1 the switch is activated.
11	–	I*C1*	Input for channel 1 which is connected to the output for channel 1 when the switch is activated.
12	–	I*C1*	Input to channel 1 which is connected to the output for channel 1 when the switch is not activated.
13	–	I*C1*	Output from channel 1.
21	–	I*C1*	Input to channel 2 which is connected to the output for channel 2 when the switch is not activated.
23		O*C1*	Output from channel 2.
: 10x*C2*+1	–	I*C1*	Input to channel C2 which is connected to the output for channel C2 when the switch is activated.
10x*C2*+2	–	I*C1*	Input to channel C2 which is connected to the output for channel C2 when the switch is not activated.
10x*C2*+3	–	O*C1*	Output from channel C2.

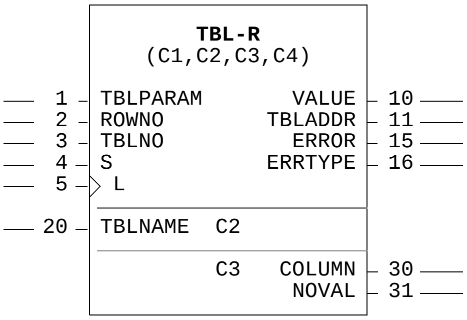
Function

When the control input ACT is 0, the data from the inputs 12 - 10x*C2*+2 are connected to the appropriate outputs. When ACT is set, data comes from the inputs 11 - 10x*C2*+1.

Read Table TBL-R

Summary

TBL-R (**T**a**BL**e **R**ead) is used to read values from the data base.



PC element TBL-R

Call TBL-R (C1, C2, C3, C4)

Parameter	Significance	Permissible values
C1	Data type	I, IL, R, B, T, TR, A
C2	Selection of symbolic search	0, 1
C3	Selection of Column reading	0, 1
C4	Number of values in the group data	1 - 255
C1=A and C3=1 is an illegal combination		

Terminal Description

No	Name	Type	Description
1	TBLPARAM	IA24	**T**a**BL**e **PARAM**eter. Connected to the TBLPARAM terminal of the TBL_PARAM data base element.
2	ROWNO	II	**ROW N**umber. Defines the row that the first value is to be read from. Allowed values 1 - 999.
3	TBLNO	II	**T**a**BL**e **N**umber. Defines which table the table values are to be read from. If TBLNO=0 the table is defined by TBLNAME. Allowed values 0 - 999.
4	S	IB	**S**. Read data from the data base on each execution cycle.
5	L	IB	**L**. Read value from the data base when the input goes from 0 to 1.
10	VALUE	O*C1*	**VALUE**. Value read from the table. Not updated if C3=1.
11	TBLADDR	OI	**T**a**BL**e **ADDR**ess. This terminal is mirrored from TBLNO. The table number computed by TBLNAME is given if TBLNO=0
15	ERROR	OB	**ERROR**. Set to 1 if there is an error.
16	ERRTYPE	OI	**ERR**or **TYPE**. Indicates error type.
20	TBLNAME	IA20	**T**a**BL**e **NAME**. Used to address a named table. Used only if TBLNO=0.

Terminal Description

No	Name	Type	Description
30	COLUMN	OG*C4C1*	**COLUMN**. Group data output containing the selected values in the column. The maximum number of values in the group data is defined by the C4 parameter.
31	NOVAL	OI	**N**umber of **VAL**ue. Indicates the number of updated values in the COLUMN group data output.

Function

TBL-R is used to read one or more values from the table data base. The element's outputs are updated when input S is set or when leading edge is detected on the L input.

The first value to be read is defined by the TBLPARAM, ROWNO and TBLNO inputs. If TBLNO=0 the TBLNAME input is used instead. TBLNAME can address the table via a symbolic name. The table number is shown on the TBLADDR output regardless of whether it is derived from TBLNO or TBLNAME.

If the COLUMN and NOVAL terminals have been selected the first position of the group data output COLUMN will contain the same value as the output terminal VALUE. The remaining positions will contain the following values in the column. The call parameter C4 determines the maximum number of values in the group data output terminal. NOVAL shows the number of values in the COLUMN terminal that have been updated.

Error Handling

If any error occurs in the element the output terminal ERROR is set to 1. Output terminal ERRTYPE shows what type of error that has occurred in the element. The following error types are possible

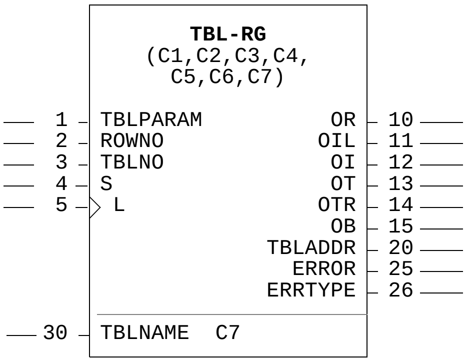
Error Types

0	No error
-999	Illegal combination of call parameters (C1=A and C3=1)
-2	Input terminal TBL PARAM is not connected
-1	Input terminal TBLNAME is not connected
1	Illegal value at input parameter ROWNO
2	Illegal value at input parameter TBLNO
3	Illegal value at both ROWNO and TBLNO terminals
4	TBLNAME not found
5	The data type in terminal TBLPARAM and the data type given by call parameter C1 is not the same

Read Table Row TBL-RG

Summary

TBL-RG (TaBLe Read Group) is used to read values from a row in the table data base.



PC element TBL-RG

Call TBL-RG (C1, C2, C3, C4, C5, C6, C7)

Parameter	Significance	Permissible values
C1	Number of values of data type R	0 - 255
C2	Number of values of data type IL	0 - 255
C3	Number of values of data type I	0 - 255
C4	Number of values of data type T	0 - 255
C5	Number of values of data type TR	0 - 255
C6	Number of values of data type TB	0 - 255
C7	Selection of symbolic search	0, 1

Terminal Description

No	Name	Type	Description
1	TBLPARAM	IA24	**T**a**BL**e **PARAM**eter. Connected to the TBLPARAM terminal of the data base element that determines the first value of the step.
2	ROWNO	II	**ROW N**umber. Defines the row that the values are to be read from. Allowed values 1 - 999.
3	TBLNO	II	**T**a**BL**e **N**umber. Defines which table the table values are to be read from. If TBLNO=0 the table is defined by TBLNAME. Allowed values 0 - 999.
4	S	IB	**S**. Read data from the data base on each execution cycle.
5	L	IB	**L**. Read value from the data base when the input goes from 0 to 1.
10	OR	OG*C1*R	**O**utput **R**eal. Group data output terminal that stores the read values of data type R.
11	OIL	OG*C2*IL	**O**utput **I**nteger **L**ong. Group data output terminal that stores the read values of data type IL.

Terminal Description

No	Name	Type	Description
12	OI	OG*C3*I	**O**utput **I**nteger. Group data output terminal that stores the read values of data type I.
13	OT	OG*C4*T	**O**utput **T**ime. Group data output terminal that stores the read values of data type T.
14	OTR	OG*C5*TR	**O**utput **T**ime **R**eal. Group data output terminal that stores the read values of data type TR.
15	OB	OG*C6*B	**O**utput **B**oolean. Group data output terminal that stores the read values of data type B.
20	TBLADDR	OI	**T**a**BL**e **ADDR**ess. This terminal is mirrored from TBLNO. The table number computed by TBLNAME is given if TBLNO=0.
25	ERROR	OB	**ERROR**. Set to 1 if there is an error.
26	ERRTYPE	OI	**ERR**or **TYPE**. Indicates error type.
30	TBLNAME	IA20	**T**a**BL**e **NAME**. Used to address a named table. Used only if TBLNO=0.

Function

TBL-RG is used to read values from a row in the table data base. The output terminals of the element are updated when the input terminal S is set or when a leading edge is detected on the L input.

The first value to be read is defined by the TBLPARAM, ROWNO and TBLNO inputs. If TBLNO=0 the TBLNAME input is used instead. TBLNAME can address the table via a symbolic name. The table number is shown on the TBLADDR output regardless of whether it is derived from TBLNO or TBLNAME.

The data types in a row may be mixed in any order. The values read from the table data base are stored in the group data output terminals of the element. The maximum number of values in these output terminals is defined with the call parameters C1 - C6. If the number of values in the output terminals are greater than the number of values in the data base the value zero (0) or "false" is stored on the unused positions. If the number of values in the output terminals is less than the number of values in the data base the reading will be stopped when the data group output terminal has been filled.

The TBL-RG (2, 2, 2, 2, 2, 2) element reads two values of each data type from the data base. The values marked x below will be read by the element.

R	R	IL	IL	I	I	T	T	TR	TR	B	B
x	x	x	x	x	x	x	x	x	x	x	x

In the above case all values are ordered by data type on the row. All values are read.

IL	R	R	T	I	I	IL	T	TR	TR	B	B
x	x	x	x	x	x	x	x	x	x	x	x

In the above case the data types are mixed. All values are read. The values are stored in the group data output terminals in the order they were read in the row.

R	R	R	IL	IL	I	I	T	T	TR	TR	B	B
x	x		x	x	x	x	x	x	x	x	x	x

This row contains a real value that will not be read.

R	R	R	I	R	I	IL	IL	IL	TR	B	B
x	x		x		x	x	x		x	x	x

This row contains more Real and IntegerLong values than the element. Only the first two R and IL values will be read. The row does not have any Time values and only one Time Real value. This means that the element's output for T and the unused TR position will be 0.

Error Handling

If any error occurs in the element the output terminal ERROR is set to 1. Output terminal ERRTYPE shows what type of error that has occurred in the element. The following error types are possible:

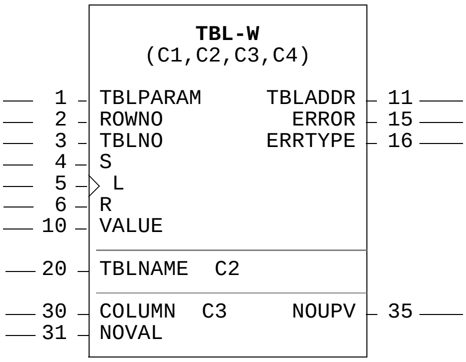
Error Types

0	No error
-2	Input terminal TBLPARAM is not connected
-1	Input terminal TBLNAME is not connected
1	Illegal value at input terminal ROWNO
2	Illegal value at input terminal TBLNO
3	Illegal value at both ROWNO and TBLNO terminals
4	TBLNAME not found

Write Table TBL–W

Summary

TBL-W (**T**a**BL**e **W**rite) is used to write data into the data base.



PC element TBL-W

Call TBL-W (C1, C2, C3, C4)

Parameter	Significance	Permissible values
C1	Data type	I, IL, R, B, T, TR, A
C2	Selection of symbolic search	0, 1
C3	Selection of column	0, 1
C4	Number of values in the group data	1 - 255
C1=A and C3=1 is an illegal combination		

Terminal Description

No	Name	Type	Description
1	TBLPARAM	IA24	**T**a**BL**e **PARAM**eter. Connected to the TBLPARAM terminal of the TBL_PARAM data base element.
2	ROWNO	II	**ROW N**umber. Defines the row that is to be updated. Allowed values 1 - 999.
3	TBLNO	II	**T**a**BL**e **N**umber. Defines which table that is to be updated. If TBLNO=0 the table is defined by TBLNAME. Allowed values 0 - 999.
4	S	IB	**S**. Write data into the data base on each execution cycle.
5	L	IB	**L**. Write data into the data base when the input goes from 0 to 1.
6	R	IB	**R**eset. Writes zero or FALSE in the indicated value in the data base. If C3=1 the number of values indicated by NOVAL are reset (only values in the current column are reset, other columns are not affected). R dominates S and L.
10	VALUE	I*C1*	**VALUE**. Value to be written into the table. Not used if C3=1.
11	TBLADDR	OI	**T**a**BL**e **ADDR**ess. This terminal is mirrored from TBLNO. The table number computed by TBLNAME is given if TBLNO=0.
15	ERROR	OB	**ERROR**. Set to 1 if there is an error.

Terminal Description

No	Name	Type	Description
16	ERRTYPE	OI	**ERR**or **TYPE**. Indicates error type.
20	TBLNAME	IA20	T**a**BL**e** **NAME**. Used to address a named table. Used only if TBLNO=0.
30	COLUMN	IG*C4C1*	**COLUMN**. The contents of the group data input is written into the data base beginning with the indicated value. The number of values to be written is determined by NOVAL. If NOVAL is greater than the remaining number of positions in the column the column will be filled and no more values will be written.
31	NOVAL	II	**N**umber of **VAL**ues. Indicates the number of values in the data base to be updated from the COLUMN group data output. Allowed values: 1 - 255.
35	NOUPV	OI	**N**umber of **UP**dated **V**alue. Indicates the number of values that have been updated in the data base from COLUMN.

Function

TBL-W is used to update values in the table data base. The outputs of the element are updated when the S input is set or when a leading edge is detected on the L input. The first value to be written is defined by the TBLPARAM, ROWNO and TBLNO inputs. If TBLNO=0 the TBLNAME input is used instead. TBLNAME can address the table via a symbolic name. The table number is shown on the TBLADDR output regardless of whether it is derived from TBLNO or TBLNAME. If the COLUMN and NOVAL terminals have been selected (C3=1) the data base will be updated with values from the group data input COLUMN. The number of values to be updated is determined by NOVAL. The update is terminated when the end of the column is reached. 0 or FALSE is written if R is set. If C3 is set the NOVAL input terminal determines the number of values to be reset. The reset is terminated when the end of the column is reached.

Error Handling

If any error occurs in the element the output terminal ERROR is set to 1. Output terminal ERRTYPE shows what type of error that has occurred in the element. The following error types are possible

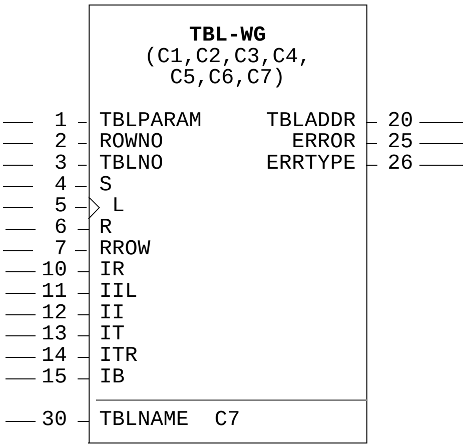
Error Types

0	No error
-999	Illegal combination of call parameters (C1=A and C3=1)
-2	Input terminal TBLPARAM is not connected
-1	Input terminal TBLNAME is not connected
1	Illegal value at input terminal ROWNO
2	Illegal value at input terminal TBLNO
3	Illegal value at both ROWNO and TBLNO terminals
4	TBLNAME not found
5	The data type in terminal TBLPARAM and the data type given by call parameter C1 is not the same
6	NOVAL is out of range

Write Table Row TBL-WG

Summary

TBL-WG (**T**a**BL**e **W**rite **G**roup) is used to update a row in the table data base.



PC element TBL-WG

Call TBL-RG (C1, C2, C3, C4, C5, C6, C7)

Parameter	Significance	Permissible values
C1	Number of values of data type R	0 - 255
C2	Number of values of data type IL	0 - 255
C3	Number of values of data type I	0 - 255
C4	Number of values of data type T	0 - 255
C5	Number of values of data type TR	0 - 255
C6	Number of values of data type T	0 - 255
C7	Selection of symbolic search	0, 1

Terminal Description

No	Name	Type	Description
1	TBLPARAM	IA24	**T**a**BL**e **PARAM**eter. Connected to the TBLPARAM terminal of the data base element that determines the first value of the row.
2	ROWNO	II	**ROW N**umber. Defines the row that is to be updated. Allowed values 1 - 999.
3	TBLNO	II	**T**a**BL**e **N**umber. Defines the table to be updated. If TBLNO=0 the table is defined by TBLNAME. Allowed values 0 - 999.
4	S	IB	**S**. Write data into the data base on each execution cycle.
5	L	IB	**L**. Write data into the data base when the input goes from 0 to 1.
6	R	IB	**R**eset. Writes zero or FALSE in the values updated via L and S. R dominates L and S.

Terminal Description

No	Name	Type	Description
7	RROW	IB	**R**eset **ROW**. Writes zero or FALS in all values in the indicated row. RROW dominates R, L and S.
10	IR	IG*C1*R	**I**nput **R**eal. Group data input terminal that gives the new values of data type R.
11	IIL	IG*C2*IL	**I**nput **I**nteger **L**ong. Group data input terminal that gives the new values of data type IL
12	II	IG*C3*I	**I**nput **I**nteger. Group data input terminal that gives the new values of data type I.
13	IT	IG*C4*T	**I**nput **T**ime. Group data input terminal that gives the new values of data type T.
14	ITR	IG*C5*TR	**I**nput **T**ime **R**eal. Group data input terminal that gives the new values of data type TR.
15	IB	IG*C6*B	**I**nput **B**oolean. Group data input terminal that gives the new values of data type B.
20	TBLADDR	OI	**T**a**BL**e **ADDR**ess. This terminal is mirrored from TBLNO. The table number computed by TBLNAME is given if TBLNO=0.
25	ERROR	OB	**ERROR**. Set to 1 if there is an error.
26	ERRTYPE	OI	**ERR**or **TYPE**. Indicates error type.
30	TBLNAME	IA20	**T**a**BL**e **NAME**. Used to address a named table. Used only if TBLNO=0.

Function

TBL-WG is used to update a row in the table data base. The selected row is identified by the TBLPARAM, ROWNO and TBLNO inputs. If TBLNO=0 the TBLNAME input is used instead. TBLNAME can address the table via a symbolic name. The table number is shown on the TBLADDR output regardless of whether it is derived from TBLNO or TBLNAME. The data types in a row may be mixed in any order.

The values to be written into the table data base are defined via the group data input terminals of the element. The maximum number of values in these input terminals is defined with the call parameters C1 - C6. If the number of values in the input terminals are greater than the number of values in the data base the value zero (0) or "false" is set on the unused positions. If the number of values in the input terminals is less than the number of values in the data base the writing will be stopped when the number of values specified in the group data input terminals have been reached.

The TBL-WG (2, 2, 2, 2, 2, 2) element updates two values of each data type into the data base. The values marked x below will be updated by the element.

R	R	IL	IL	I	I	T	T	TR	TR	B	B
x	x	x	x	x	x	x	x	x	x	x	x

In the above case all values are ordered by data type on the row. All values are updated.

IL	R	T	R	I	I	IL	T	TR	TR	B	B
x	x	x	x	x	x	x	x	x	x	x	x

In the above case the data types are mixed. All values will be updated.

R	R	R	IL	IL	I	I	T	T	TR	TR	B	B
x	x		x	x	x	x	x	x	x	x	x	x

This row contains a real value that will not be read.

R	R	R	R	I	I	IL	IL	IL	TR	B	B
x	x			x	x	x	x		x	x	x

This row contains more Real and IntegerLong values than the element. Only the first two R and IL values will be updated. The row does not have any Time values and only one Time Real value. This means that the Time Real value will be updated with the first value from the group data input terminal of this type.

R	R	R	I	I	R	IL	IL	B	TR	B	B	B
0	0	0	0	0	0	0	0	F	0	F	F	F

If RROW is set all values in the row will be reset.

Error Handling

If any error occurs in the element the output terminal ERROR is set to 1. Output terminal ERRTYPE shows what type of error that has occurred in the element. The following error types are possible

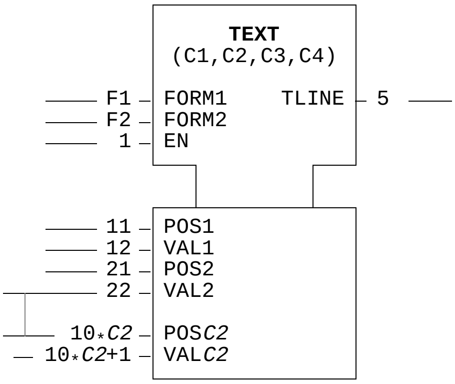
Error Types

0	No error
-2	Input terminal TBLPARAM is not connected
-1	Input terminal TBLNAME is not connected
1	Illegal value at input terminal ROWNO
2	Illegal value at input terminal TBLNO
3	Illegal value at both ROWNO and TBLNO terminals
4	TBLNAME not found

Text Line TEXT

Summary

TEXT is used to assemble a text line of up to 8 subtexts.



PC element TEXT

Call TEXT (C1, C2, C3, C4)

Parameter	Significance	Permissible values
C1	Data type	B, I, IL, T, TR, R, A
C2	Number of sub components in the text	1 - 8
C3	Number of characters if data type A is used	1 - 80
C4	Number of characters for the output TLINE	1 - 80

Terminal Description

No	Name	Type	Description
F1	FORM1	FI	**FORM**at **1**. Format specification for total length with the data types I, IL, R and TR.
F2	FORM2	FI	**FORM**at **2**. Format specification for the decimal part with the data types R and TR. FORM2=–6..+13.
1	EN	IB	**EN**able. Input for activation of execution. The element is executed if EN = 1.
5	TLINE	OA*C4*	**T**ext **LINE**. Output for generated text.
11	POS1	II	**POS**ition **1**. Specifies in which position in the text field 1 is to begin.
12	VAL1	I*C1*	**VAL**ue **1**. Value to be inserted into the text.
21	POS2	II	**POS**ition **2**. Specifies in which position in the text field 2 is to begin.
22	VAL2	I*C1*	**VAL**ue **2**. Value which is to be inserted into the text.
: 10x*C2*+1	POS*C2*	II	**POS**ition ***C2***. Specifies in which position in the text the last field is to begin.
10x*C2*+2	VAL*C2*	I*C1*	**VAL**ue ***C2***. Value which is to be inserted into the text.

Function

TEXT is used to assemble a text line of up to 8 fields. These can be of type B, I, IL, T, TR, R or A according to the call parameter C1. The call parameter C2 gives the number of fields to be assembled. If the data type is A, the call parameter C3 indicates how many characters are included in the text.

Location of Fields

The value for each field is given at the input VAL. POS specifies in which position in the line (1 - C4) the first position of the field is to be placed.

Handling of Integers (I) and Long Integers (IL)

The length of the field is given by the function parameter FORM1. The value is written right adjusted within the field (POS - (POS+FORM1–1)). If FORM1=0, the field length for I defaults to 6 and for IL to 11. If a (long) integer cannot be accommodated within the field length, the value is replaced with ”*” in the entire field.

Handling of Time (T)

For time (T) the field length is always 10. Time is written out in the form hours:minutes:seconds.decimal. The decimal part is specified to 1 digit.

Handling of Real Numbers (R) and Real Time (TR)

For these data types, FORM1 gives the total length whereas FORM2 gives the number of digits in the decimal part of the number. If the number is too small, it is written out in exponential form with the number of characters accommodated in the field. If FORM1=FORM2=0 a field of 15 characters with 3 characters for the decimal part of the number is used. The decimal point and the sign are included in the total field length. If the data type real is required to be written out in exponential form, FORM1 is set to the total field length and FORM2 is set to the negative of the number of characters in the exponent.

Handling of Text (A) and Boolean (B)

The field size for a text is the size of the text concerned. If '$DATE' or '$TIME' is given at the input the current date and time are written in the form year:month:day and hour:minute:second.decimal respectively. If a character sequence is to be repeated on a line, a text consisting of a dollar sign ($) followed by a number and the text which is to be repeated is entered. For example, '$60*' gives a line of 60 '*':

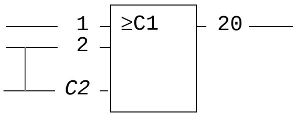
****

Boolean occupies 1 character.

Threshold Function THRESH-L

Summary

THRESH-L (**THRESH**old **- L**ogic) is used to determine when more than, or equal to, a given number of Boolean signals are set (to 1).



PC element THRESH-L

Call THRESH-L (C1, C2)

Parameter	Significance	Permissible values
C1	Threshold value	1 - C2
C2	Number of inputs	1 - 19

Terminal Description

No	Name	Type	Description
1	–	IB	Input for the first Boolean value.
2	–	IB	Input for the second Boolean value.
: *C2*	–	IB	Input for the last Boolean value.
20	–	OB	Output which is set to 1 if C1 or more of the inputs 1 - C2 are set.

Function

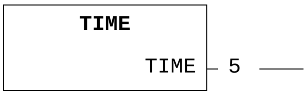
The TRESH-L element is used to determine if more than, or equal to, a specified number of Boolean values in a group are set to 1. If so, the output (20) is set.

Time Reading

TIME

Summary

TIME is used for reading the current system time.



PC element TIME

Call TIME

Terminal Description

No	Name	Type	Description
5	TIME	OT	Output for system time.

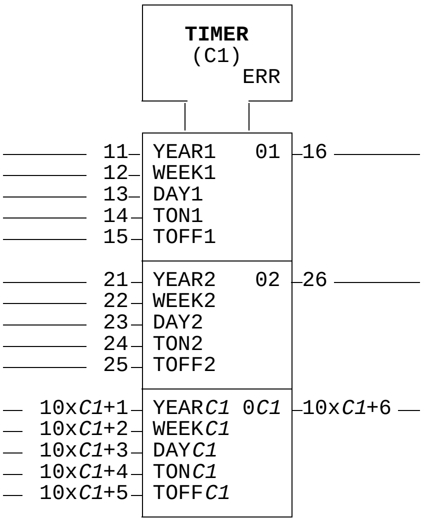
Function

The current system time is obtained at the output TIME. This can be used, for example, for printouts via TEXT and PRINT elements. The element can be used for calculations or for control of multiplexers by converting the system time to integers with the help of the CONV element.

Time Selector TIMER

Summary

TIMER is used to generate a pulse with a duration determined by two times in the same day (24 hour).



PC element TIMER

Call TIMER (C1)

Parameter	Significance	Permissible values
C1	Number of times	1 - 9

Terminal Description

No	Name	Type	Description
5	ERR	OB	**ERR**or. Error signal which is set to one when any of the inputs have values outside the permitted time range.
11	YEAR1	II	Input which specifies which year the pulse is to be given from the first time element. The value 0 means a pulse is to be given each year.
12	WEEK1	II	Input which specifies which week the pulse is to be given from the first time element. The value 0 means a pulse is to be given each week.
13	DAY1	II	Input which specifies which day of the week the pulse is to be given from the first time element. The value 0 means a pulse is to be given each day.
14	TON1	IT	**T**ime **ON 1**. Input which specifies at which time the pulse from the first time element is to begin.
15	TOFF1	IT	**T**ime **OFF 1**. Input which specifies at which time the pulse from the first time element is to end.
16	O1	OB	**O**utput **1**. Output at which a pulse appears between the times specified at the inputs TON1 and TOFF1.
21	YEAR2	II	Input which specifies which year the pulse is to be given from the second time element. The value 0 means a pulse is to be given each year.

Terminal Description

No	Name	Type	Description
22	WEEK2	II	Input which specifies which week the pulse is to be given from the second time element. The value 0 means a pulse is to be given each week.
23	DAY2	II	Input which specifies which day of the week the pulse is to be given from the second time element. The value 0 means a pulse is to be given each day.
24	TON2	IT	**T**ime **ON 2**. Input which specifies at which time the pulse from the second time element is to begin.
25	TOFF2	IT	**T**ime **OFF 2**. Input which specifies at which time the pulse from the second time element is to end.
26	O2	OB	**O**utput **2**. Output at which a pulse appears between the times specified at the inputs TON2 and TOFF2.
: 10x*C1*+1	YEAR*C1*	II	Input which specifies which year the pulse is to be given from the last time element. The value 0 means a pulse is to be given each year.
10x*C1*+2	WEEK*C1*	II	Input which specifies which week the pulse is to be given from the last time element. The value 0 means a pulse is to be given each week.
10x*C1*+3	DAY*C1*	II	Input which specifies which day of the week the pulse is to be given from the last time element. The value 0 means a pulse is to be given each day.
10x*C1*+4	TON*C1*	IT	**T**ime **ON** *C1*. Input which specifies at which time the pulse from the last time element is to begin.
10x*C1*+5	TOFF*C1*	IT	**T**ime **OFF** *C1*. Input which specifies at which time the pulse from the last time element is to end.
10x*C1*+6	O*C1*	OB	**O**utput ***C1***. Output at which a pulse appears between the times specified at the inputs TON*C1* and TOFF*C1*.

Function

TIMER is used to produce a pulse between two times within one (24 hours) day. The day is specified by year, week number and day number within the week. The duration of the pulse is determined by the inputs TON and TOFF. The generated pulse can be used to start required functions.

YEAR Input

The year during which the pulse is to be obtained is specified at the inputs YEAR1 - YEAR*C1*. If the year is given as 0, a pulse is given each year. In addition to 0, years 1980 to 2099 can be given.

WEEK Input

The week during which the pulse is to be obtained is specified at the inputs WEEK1 - WEEK*C1*. If the week is given as 0, a pulse is given each week. In addition to 0, week numbers 1 to 53 can be given. If week 53 is specified in a year that does not have a week 53, no pulse is produced.

DAY Input

The day of the week during which the pulse is to be obtained is specified at the inputs DAY1 - DAY*C1*. If the day is given as 0, a pulse is given each day of the week. In addition to 0, day numbers 1 to 7 can be given. Day 1 refers to Monday and day 7 is Sunday.

Rules for Week and Day Numbering

Weeks and days are numbered in accordance with the ISO standard. Monday is considered to be the first day of the week. The first week which contains at least 4 days is considered to be the first week of the year. When days are referred to by year-week-day, the change of year is on week 1, day 1.

Example: 1 Jan 1995 is given as 1994 52 6

TON and TOFF Inputs

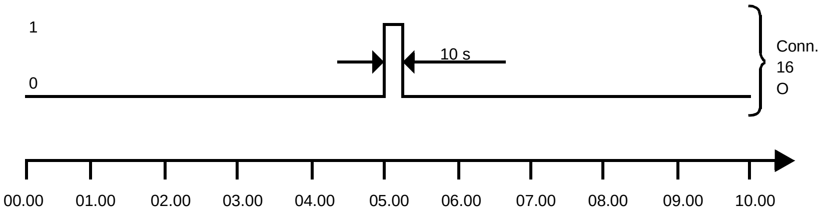
At which time a pulse is to begin or end respectively is given at the inputs TON and TOFF. TON and TOFF must be within the same 24 hour period, that is, TON<TOFF. Time specifications can vary from 00:00:00.000 to 23:59:59.999. For repetition within a 24 hour period, for example, each hour, external elements are required.

Examples of Selection of Day

YEAR	WEEK	DAY	Significance
0	0	0	Each day.
1980 - 2099	0	0	Each day during a certain year.
1980 - 2099	1 - 53	0	Each day during a certain week during a certain year.
1980 - 2099	0	1 - 7	A certain day each week during a certain year.
0	1 - 53	0	Each day during a certain week every year.
0	0	1 - 7	A certain day of each week, every year.
0	1 - 53	1 - 7	A certain day of each year.
1980 - 2099	1 - 53	1 - 7	A particular day.

Supervision

If any input has a value outside the permitted range, the error signal ERR is set (to 1).



Time pulse diagram

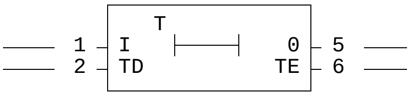
Time Delay On/ Time Delay Off

TON/ TOFF

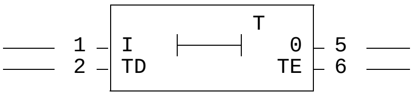
Summary

Time delay on TON (**T**imer **ON**-delay) and time delay off TOFF (**T**imer **OFF**-delay) of Boolean variables for use in connection with combined expressions.

Call TON TOFF



PC element TON



PC element TOFF

Terminal Description

No	Name	Type	Description
1	I	IB	**I**nput for start of time delay.
2	TD	IT	**T**ime **D**elay. Input for preset time. Max 23h 59m 59.999s.
5	O	OB	**O**utput. Output which is set (TON) or reset (TOFF) when the preset time has elapsed.
6	TE	OT	**T**ime **E**lapsed. Output which specifies how long I has been set (TON) or reset (TOFF). When the preset time (TD) has elapsed, TE stops.

Function

TON

The input variable to the input I is obtained delayed at output O when the input variable changes from 0 to 1. The output signal returns immediately the input variable changes from 1 to 0.

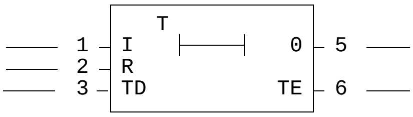
TOFF

The input variable to the input I is obtained delayed at output O when the input variable changes from 1 to 0. The output signal is set immediately the input variable changes from 0 to 1.

Time Delay On with Retention TON-RET

Summary

TON-RET (**T**ime delay **ON - RET**entive) switch-on delay with accumulating time measurement of Boolean variables for use in connection with combined expressions. TON-RET is used for, for example, operational time monitoring.



PC element TON-RET

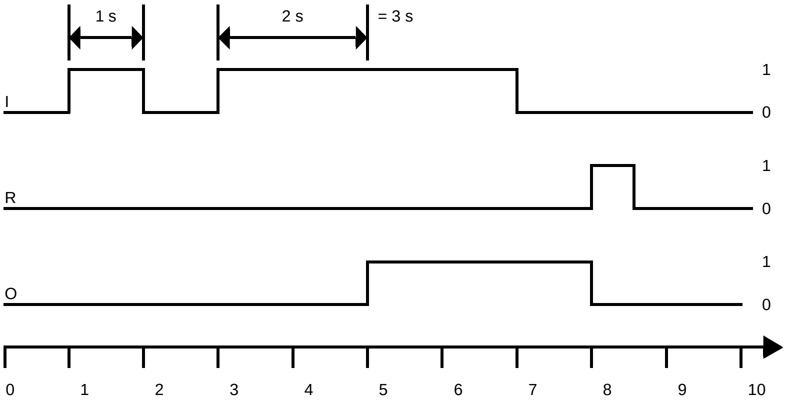
Call TON-RET

Terminal Description

No	Name	Type	Description
1	I	IB	**I**nput for start of time delay when the input changes from 0 to 1. The time circuit maintains its status even when the input returns to 0. **O**utput which is set when the preset time has elapsed.
2	R	IB	**R**eset. Input which resets the timer. R must be reset (to 0) for the timer to function.
3	TD	IT	**T**ime **D**elay. Input for preset time, that is, the time the input I must be set (to 1). Max 23h 59m 59s.
5	O	OB	**O**utput which is set when the preset time has elapsed.
6	TE	OT	**T**ime **E**lapsed. Output which indicates the time during which I has been set to 1.

Function

The input variable to the input I is obtained delayed at the output O when the input variable changes from 0 to 1 in accordance with the time pulse diagram in figure 2. If the input variable returns to 0, the time which has elapsed remains in the timer and when the variable returns to 1, the time continues from the value which applied when the variable went to 0. The input R must be reset (to 0) when the timer functions. If R is set (to 1) both outputs O and TE are reset.

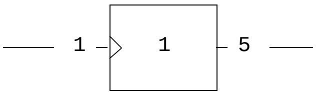


Example of time diagram with preset time 3 s

Trigger TRIGG

Summary

The triggering element TRIGG is used for, for example, reducing impulse times at the start of automatic procedures and for calculating functions.



PC element TRIGG

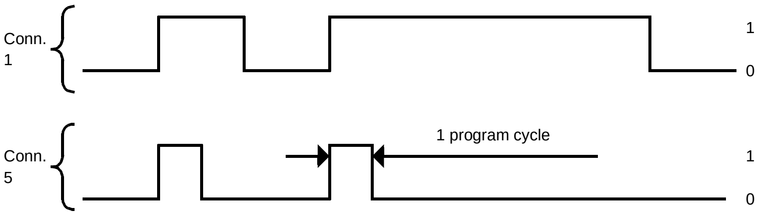
Call TRIGG

Terminal Description

No	Name	Type	Description
1	–	IB	Input for start of trigger pulse.
5	–	OB	Output for trigger pulse.

Function

When the input signal is set (to 1) the output signal is also set. The output signal is cleared during the next program cycle, irrespective of the value of the input signal. For the output signal to be set, the input signal must have been 0 during one program cycle. The output signal from the element may only be used within its own execution unit as detection of it is otherwise not certain.



Time diagram

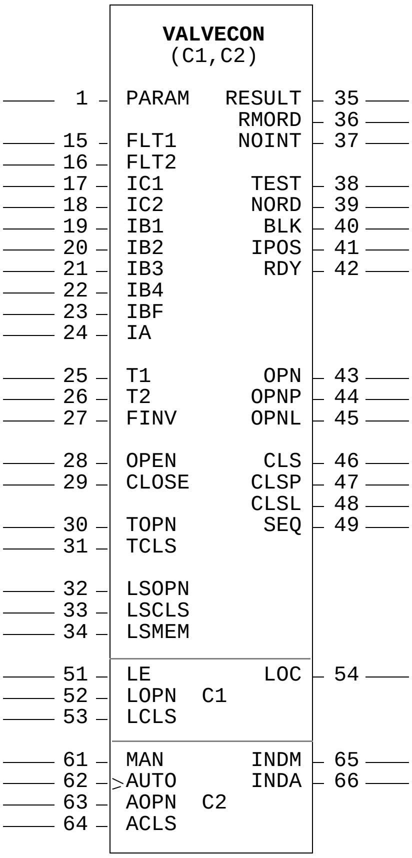
NOTE

If input signal is set at program startup (DICONFIG or DBL), the output is also set during the first program cycle.

Valve Controller VALVECON

Summary

The PC element VALVECON forms part of the functional unit VALVECON (VALVE CONtroller). The element is used with the data base element VALVECON for controlling valves. Further information can be found in the manual *Functional Units part 6, MOTCON, VALVECON*.



PC element VALVECON

Call VALVECON (C1, C2)

The element consists of a basic part which forms a complete basic valve controller. This function may be extended with optional extra functions with the call parameters C1 and C2 according to the table below.

Parameter	Significance	Permissible values
C1	Selection of function for local control	0 - 1
C2	Selection of function for control mode AUTO	0 - 1

Terminal Description

No	Name	Type	Description
1	PARAM	IA32	**PARAM**eter. Parameters from the object data base element VALVECON, terminal PC_PARAM
15	FLT1	IB	**F**au**LT 1**. Input for external error 1.
16	FLT2	IB	**F**au**LT 2**. Input for external error 2.
17	IC1	IB	**I**nterlock **C1**. Operation interlock (not deblockable).
18	IC2	IB	**I**nterlock **C2**. Operation interlock (not deblockable).
19	IB1	IB	**I**nterlock **B1**. Operation interlock$^{(1)(2)}$.
20	IB2	IB	**I**nterlock **B2**. Operation interlock(2).
21	IB3	IB	**I**nterlock **B3**. Start/Operation interlock(1)(2).
22	IB4	IB	**I**nterlock **B4**. Start/Operation interlock.
23	IBF	IB	**I**nterlock **B F**unction. Control input for IB3 and IB4. If IBF is set to 1 IB3 and IB4 works as start interlocks.
24	IA	IB	**I**nterlock **A**. Operation interlock(2).
25	T1	IT	**T**ime **1**. Input for time for OPEN/CLOSE pulse.
26	T2	IT	**T**ime **2**. Input for maximum travel time.
27	FINV	IB	**F**unction **INV**ersion. Input for home position. FINV=0 gives home position closed. FINV=1 gives home position open.
28	OPEN	IB	Input for open order.
29	CLOSE	IB	Input for close order.
30	TOPN	IB	**T**est **OP**e**N**. Input for open order in TEST mode.
31	TCLS	IB	**T**est **CL**o**S**e. Input for close order in TEST mode.
32	LSOPN	IB	**L**imit **S**witch **OP**e**N**. Input for limit position open.
33	LSCLS	IB	**L**imit **S**witch **CL**o**S**ed. Input for limit position closed.
34	LSMEM	IB	**L**imit **S**witch **MEM**ory. The function is not implemented.
35	RESULT	OA12	Output for results to the data base element VALVECON, terminal PC_RES.
36	RMORD	OI	**R**eset **M**anual **ORD**er. Output for resetting operation order in the object data base element VALVECON, terminal MORD.
37	NOINT	OB	**NO INT**erlock. Output which is set to 1 when the object is not interlocked and no external fault reasons (FLT1 or FLT2) exist.
38	TEST	OB	Output for indication of point of control TEST.
39	NORD	OB	**N**ew **ORD**er. Output for open/close order (pulse).
40	BLK	OB	**BL**oc**K**ed. Output which is set to 1 when the point of control is STANDBY.

Terminal Description

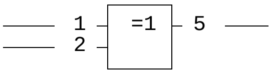
No	Name	Type	Description
41	IPOS	OB	Intermediate **POS**ition. Output for indication of intermediate position.
42	RDY	OB	**R**ea**DY**. Output for ready indication.
43	OPN	OB	**OP**e**N**. Continuous output for OPEN.
44	OPNP	OB	**OP**e**N** **P**ulse. Pulse output for OPEN (pulse during time T1).
45	OPNL	OB	**OP**e**N** **L**imit. Output for OPEN which is deactivated when LSOPN has been reached.
46	CLS	OB	**CL**o**S**e. Continuous output for CLOSE.
47	CLSP	OB	**CL**o**S**e **P**ulse. Pulse output for CLOSE (pulse during time T1).
48	CLSL	OB	**CL**o**S**e **L**imit. Output for CLOSE which is deactivated when LSCLS has been reached.
49	SEQ	OB	Indication of point of control **SEQ**uence.
51(3)	LE	IB	**L**ocal **E**nable. Input for selecting point of control LOCAL.
52(3)	LOPN	IB	**L**ocal **OP**e**N** order. Input for local open order.
53(3)	LCLS	IB	**L**ocal **CL**o**S**e order. Input for local close order.
54(3)	LOC	OB	**LOC**al. Indication of point of control LOCAL.
61(4)	MAN	IB	**MAN**ual. Input for switching to control mode MAN.
62(4)	AUTO	IB	**AUTO**matic. Input for switching to control mode AUTO. Dynamic input.
63(4)	AOPN	IB	**A**uto **OP**e**N**. Input for auto open order.
64(4)	ACLS	IB	**A**uto **CL**o**S**e. Input for auto close order.
65(4)	INDM	OB	**IND**ication **M**anual. Output for indication of control mode MAN
66(4)	INDA	OB	**IND**ication **A**uto. Output for indication of control mode AUTO.

(1) May be disabled via operator dialog
(2) Disabled in operation mode TEST
(3) These terminals are selected with the call parameter C1.
(4) These terminals are selected with the call parameter C2.

Exclusive Or Gate XOR

Summary

XOR (e**X**clusive **OR**) is used to generate combined expressions with Boolean variables.



PC element XOR

Call XOR

Terminal Description

No	Name	Type	Description
1	–	IB	Input
2	–	IB	Input
5	–	OB	Output

Function

The output signal from the XOR element is 1 if the input signals are different and 0 if they are equal, see the table below. By inverting the output of the XOR element, a 1 is obtained if the signals are equal.

Table Input and Output Signals

1	2	5
0	0	0
0	1	1
1	0	1
1	1	0

3BSE 002 418R601

January, 2000